



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV RADIOELEKTRONIKY

DEPARTMENT OF RADIO ELECTRONICS

INTELIGENTNÍ SYSTÉM PRO SPRÁVU AKVÁRIA

SMART AQUARIUM MANAGEMENT SYSTEM

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Libor Musil

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Lucie Hudcová, Ph.D.

BRNO 2018

Bakalářská práce

bakalářský studijní obor **Elektronika a sdělovací technika**

Ústav radioelektroniky

Student: Libor Musil

ID: 186145

Ročník: 3

Akademický rok: 2017/18

NÁZEV TÉMATU:

Inteligentní systém pro správu akvária

POKYNY PRO VYPRACOVÁNÍ:

Navrhnete a realizujete inteligentní systém pro správu akvária. Řídicí systém bude založen na platformě i.Mx a operačním systémem reálného času QNX. Přístroj bude na základě údajů z PH metru řídit zásobování vody oxidem uhličitým. Údaje získané ze senzorů budou poskytovány na grafickém dotykovém displeji a prostřednictvím síťového rozhraní.

Seznamte se s platformou i.Mx v kombinaci s operačním systémem QNX, možnostmi použití a vývojem aplikací pro tuto platformu. Prozkoumejte možnosti měření pH kapalin a vyberte vhodný senzor pro měření PH akvarijní vody a elektroventil pro ovládání plynu. Sestrojte vhodné hardwarové rozhraní mezi řídicí deskou, která umožní sběr dat z připojených senzorů (pH metr, čidlo teploty vody, čidlo okolního osvětlení) a řízení ovladačů (ventil na CO₂, PWM modulátor pro řízení LED osvětlení).

Vytvořte aplikaci pro operační systém QNX, která bude běžet na řídicí desce a bude se starat o řízení sycení akvarijní vody oxidem uhličitým. Umožněte vzdálenou správu zařízení pomocí webového rozhraní. Součástí práce je rovněž návrh grafického uživatelského rozhraní pro dotykový displej.

DOPORUČENÁ LITERATURA:

[1] QNX® Software Development Platform 7.0. QNX Software Development Platform [online].

Dostupné z: <http://www.qnx.com/developers/docs/7.0.0/#com.qnx.doc.qnxsdv.nav/topic/bookset.html>

[2] CHROBOCZEK, Martin. Grafická uživatelská rozhraní v Qt a C++: [plně kompatibilní s Qt 5]. Brno: Computer Press, 2013. ISBN 9788025141243.

Termín zadání: 5. 2. 2018

Termín odevzdání: 24. 5. 2018

Vedoucí práce: Ing. Lucie Hudcová, Ph.D.

Konzultant: Ing. Lukáš Janík, NXP



prof. Ing. Tomáš Kratochvíl, Ph.D.
předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Abstrakt

Tato práce se zabývá návrhem inteligentního systému pro správu akvária. Systém je založený na platformě i.Mx8DV s grafickým výstupem na dotykový displej. V práci jsou diskutovány způsoby měření a regulace veličin kritických pro správný chod akvária. Dále je v práci popsána platforma i.Mx8DV a operační systém Neutrino. Práce řeší též výběr vhodných čidel a návrh desky plošných spojů k propojení platformy i.Mx8DV s vybranými čidly. Jsou zde popsány principy komunikace po síti a možnosti implementace webového serveru. Závěrem práce je popsána vytvořená aplikace.

Klíčová slova

Akvaristika, měření teploty, měření pH, i.Mx8DV, Neutrino, TCP server, Qt.

Abstract

This bachelor thesis deals with the design of an intelligent aquarium management system. The system is going to be based on the i.Mx8DV platform with graphical output on the touch screen. In the thesis, the methods of measuring and controlling the quantities, that are critical for the proper operation of the aquarium, are discussed. Also, the i.Mx8DV platform and the Neutrino operating system are described. This thesis solves the selection of suitable sensors and design of printed circuit board for connection of the i.Mx8DV platform with selected sensors. The principles of network communication and web server implementation options are described. In the end, the application is described.

Keywords

Aquaristics, temperature measurement, pH measurement, i.Mx8DV, Neutrino, TCP server, Qt.

MUSIL, L. *Inteligentní systém pro správu akvária*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2018. 63 s., 3 s. příloh. Vedoucí bakalářské práce Ing. Lucie Hudcová, Ph.D.

Prohlášení

Prohlašuji, že svou bakalářskou práci na téma Inteligentní systém pro správu akvária jsem vypracoval samostatně pod vedením vedoucí bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **22. května 2018**

.....
podpis autora

Poděkování

Rád bych zde poděkoval vedoucí bakalářské práce Ing. Lucii Hudcové, Ph.D. a konzultantovi Ing. Lukášovi Janíkovi z firmy NXP za jejich ochotu, trpělivost, čas a rady, které mi věnovali při řešení této bakalářské práce.

V Brně dne: **22. května 2018**

.....
podpis autora

OBSAH

Seznam symbolů, veličin a zkratek	x
Seznam obrázků	xi
Seznam tabulek	xiii
Úvod	1
1 Podmínky pro život v akváriu	2
1.1 Teplota vody	3
1.2 pH vody.....	5
1.3 Intenzita osvětlení	6
2 Použitá platforma, operační systém a Vývojové prostředí	7
2.1 i.Mx8.....	7
2.1.1 System on chip.....	7
2.1.2 i.Mx8DV	7
2.2 Operační systém.....	8
2.2.1 Operační systém reálného času.....	9
2.2.2 Monolitické jádro a mikrojádro	9
2.2.3 QNX Neutrino.....	10
2.3 Framework Qt	10
2.3.1 Licence Qt.....	11
2.3.2 Signály a sloty.....	11
2.3.3 Základní třídy Qt.....	12
2.4 Vývojové prostředí Qt creator	13
3 Návrh HW rozhraní	14
3.1 Teplotní čidlo	14
3.1.1 Termočlánek	14
3.1.2 Termistor.....	15
3.1.3 Integrovaný obvod LM35	17
3.2 Topné těleso	18
3.3 pH senzor	18
3.4 Dávkování CO ₂	19

3.5	Senzor intenzity světla	20
3.6	Osvětlení	22
3.6.1	PWM modulace	22
3.7	A/D převodník	22
3.7.1	Pojmy spojené s A/D převodníkem	23
3.7.2	Základní typy A/D převodníků podle způsobu převodu.....	23
3.7.3	Výběr A/D převodníku	24
4	Komunikace po síti	25
4.1	Ethernet	25
4.1.1	Klasický Ethernet.....	25
4.1.2	Fast Ethernet	25
4.1.3	Gigabit Ethernet.....	25
4.2	Síťový model OSI	26
4.2.1	Vrstvy OSI	26
4.3	Síťový model TCP/IP	27
4.4	TCP	28
4.4.1	TCP Hlavička.....	28
4.4.2	Komunikace protokolem TCP	29
4.5	Socket.....	29
4.6	HTTP protokol.....	30
4.6.1	Požadavek klienta	30
4.6.2	Odpověď serveru	31
4.7	HTML	31
5	Softwarové řešení	32
5.1	Možná řešení.....	32
5.1.1	Vzdálený server	32
5.1.2	Server mimo aplikaci	32
5.1.3	Server v aplikaci	33
5.2	Architektura vytvořeného systému	34
5.3	Realizace	35
5.3.1	Přehled signálů a slotů	35
5.3.2	Algoritmus komunikace klienta se serverem.....	36
5.3.3	Algoritmus změny nastavení	37

5.4	Popis vytvořeného grafického uživatelského rozhraní	38
5.4.1	Hlavní grafické okno	38
5.4.2	Obrazovka nastavení.....	38
5.4.3	Kalibrace sondy	39
5.5	Webové rozhraní.....	40
6	Testování zařízení	42
6.1	Naměřená data	42
6.1.1	Bez dodávání CO ₂	42
6.1.2	S dodáváním CO ₂	42
6.1.3	Konstantní hodnota pH	43
6.2	Možná vylepšení	44
7	Závěr	46
	Literatura	47
A	Návrh zařízení	51
A.1	Obvodové zapojení	51
A.2	Deska plošného spoje řídicího modulu – bottom (strana spojů).....	52
B	Seznam součástek	53

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

t	čas
q	kvantizační krok
N	počet bitů
k	citlivost
GPIO	General-purpose input/output
PWM	Pulse Width Modulation
GPS	Global Positioning System
HW	Hardware
SW	Software
RTOS	Real-time operating system
PTC	Positive temperature coefficient
NTC	Negative temperature coefficient
LED	Light-Emitting Diode
SDK	Software development kit
IDE	Integrated Development Environment
CGI	Common Gateway Interface
GUI	Graphical User Interface
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
IEEE	Institute of Electrical and Electronics Engineers
UTP	Unshielded Twisted Pair
STP	Shielded Twisted Pair
OSI	Open Systems Interconnection
ISO	International Organization for Standardization
TCP	Transmission Control Protocol
IP	Internet Protocol
HTTP	Hypertext Transfer Protocol
URL	Uniform Resource Locator
HTML	HyperText Markup Language
pH	potential of hydrogen

SEZNAM OBRÁZKŮ

Obrázek 1: Zobrazení koloběhu látek v akváriu [1]	2
Obrázek 2: Zobrazení Tabulky 1: Přehled rozsahů teplot pro živočichy a rostliny v akváriu	4
Obrázek 3: Topné těleso [3] Obrázek 4: Topný kabel [4]	5
Obrázek 5: Rozsah ideálních hodnot pH pro organismy v akváriu	6
Obrázek 6: Popis i.Mx8DV	8
Obrázek 7: Zobrazení vztahu mezi operačním systémem, HW, aplikačním SW a uživatelem [12]	8
Obrázek 8: Rozdíl mezi monolitickým jádrem a mikrojádro [11].....	10
Obrázek 9: Znázornění signálů a slotů [33].....	12
Obrázek 10: Zobrazení Qt Creatoru.....	13
Obrázek 11: charakteristiky termočlánků [19]	15
Obrázek 12: Závislost odporu PT100 na teplotě [17].....	16
Obrázek 13: Závislost výstupního napětí na teplotě čidla LM35 [20]	17
Obrázek 14: Ukázka profesionálního pH metr PH 100 ATC [28]	18
Obrázek 15: Modul pro měření pH [14]	19
Obrázek 16: Řetězec dodání CO2.....	19
Obrázek 17: Zobrazení tlakové lahve s ventilem [29].....	20
Obrázek 18: Lux-Ohmová charakteristika fotorezistoru GL5516 [18]	21
Obrázek 19: Zobrazení průběhu PWM [38]	22
Obrázek 20: Porovnání modelů OSI a TCP/IP	26
Obrázek 21: TCP hlavička.....	29
Obrázek 22: Zobrazení principu socketů na bázi klient-server	30
Obrázek 23: Princip komunikace přes vzdálený server.....	32
Obrázek 24: Princip výměny informací pomocí souborů.....	33
Obrázek 25: Princip serveru v aplikaci.....	33
Obrázek 26: Zobrazení architektury aplikace	34
Obrázek 27: Zobrazení signálů a slotů v aplikaci.....	35
Obrázek 28: Algoritmus obsluhy požadavku klienta.....	36
Obrázek 29: Algoritmus pro zpracování dat při jejich změně.....	37
Obrázek 30: Úvodní obrazovka GUI.....	38
Obrázek 31: Obrazovka s nastavením	39
Obrázek 32: Kalibrace sondy krok 1	39

Obrázek 33: Kalibrace sondy krok 2	40
Obrázek 34: Zobrazení webového rozhraní.....	41
Obrázek 35: Zobrazení průběhu hodnot pH vody během dne bez dodávání CO ₂	42
Obrázek 36: Zobrazení hodnot pH vody při dodávání CO ₂	43
Obrázek 37: Zobrazení hodnot pH vody.....	44
Obrázek 38: Přepět'ová ochrana signálové cesty	45

SEZNAM TABULEK

Tabulka 1: Přehled rozsahů teplot pro živočichy a rostliny v akváriu [2]	3
Tabulka 2: Teplotní závislost PT100 [17]	16

ÚVOD

Akvaristika je oblíbená zájmová činnost s miliony příznivců po celém světě. Akvaristé musejí vynakládat velké množství úsilí k tomu, aby zajistili podmínky, které věrně napodobují přirozené prostředí chovaných živočichů a rostlin nebo alespoň k tomu, aby toto prostředí bylo slučitelné s chovem v akváriu. Za úspěch v akvaristice se většinou považuje úspěšný odchov mláďat či množení rostlin, neboť jednotlivé druhy mívají svoje velice specifické podmínky pro rozmnožování.

Tohoto úspěchu se dosahuje sledováním a úpravou mnoha parametrů vody jako je například teplota či hodnota pH, dodáváním potřebných živin, odstraňování znečištění produkovaného živočichy a dalšími postupy.

Tato práce se zabývá vývojem systému správy akvária. Systém má být založený na vývojové platformě i.Mx8DV a operačním systémem reálného času QNX Neutrino. Systém bude plnit dvě základní funkce – měření veličin ovlivňující kvalitu vody v akváriu a regulaci těchto veličin. Systém bude ovládán přes dotykový displej umístěný na vývojové desce a dále pomocí síťového připojení.

Jedním z cílů této práce je analyzovat požadavky důležité pro úspěšný provoz akvária, tedy zejména zajištění dobrých podmínek pro živočichy a rostliny v něm, a možnosti jejich automatizace. Vytvoření vhodného hardwarového rozhraní mezi řídicí deskou a deskou s akčními členy a čidly.

Dále si tato práce klade za cíl navrhnout a vytvořit aplikaci pro systém správy akvária, která bude dostupná pomocí dotykového displeje a přes webové rozhraní.

1 PODMÍNKY PRO ŽIVOT V AKVÁRIU

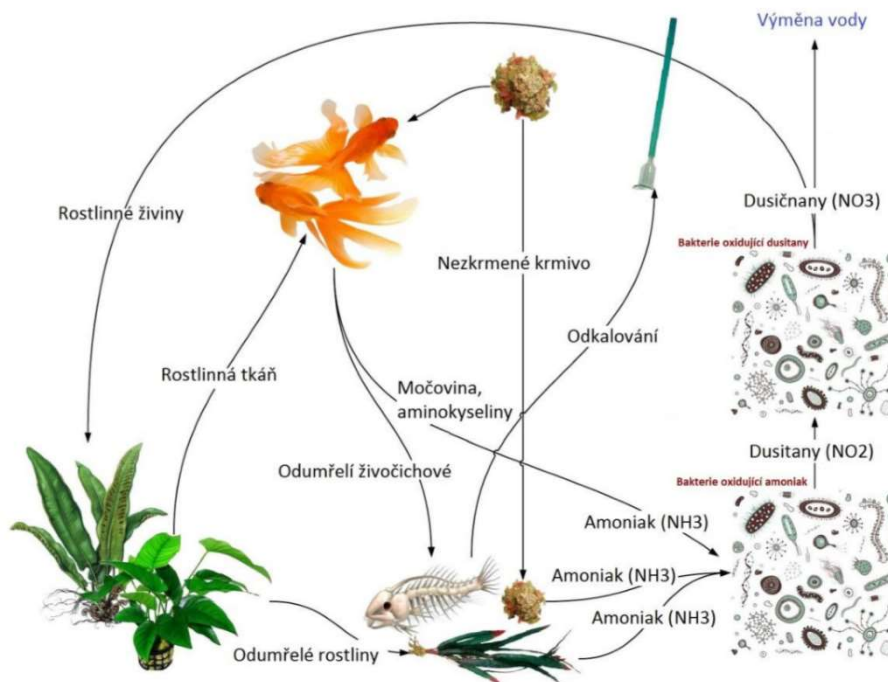
Nalítím vody do nádoby se z ní ještě nestává akvárium. Bez biologické rovnováhy a vnějších zásahů by se voda zkažila a časem by se její kvalita stala neslučitelná s přežitím živočichů a rostlin v ní.

Biologická rovnováha je stav, kdy všechny složky nutné k transformaci (koloběhu) organických a anorganických látek tvořených organismy jsou přítomné, fungují a vzájemně se doplňují. Jedná se o proces, kde organismy spotřebovávají kyslík a organické látky. Přitom vylučují oxid uhličitý a odpadní látky, které se společně s odumřelými zbytky rostlin a živočichů mění na dusíkaté sloučeniny. Tyto sloučeniny jsou při vyšších množstvích pro většinu živočichů toxické. Dané látky jsou dále přeměňovány nitrifikačními bakteriemi žijícími ve filtru na méně nebezpečné látky. Část těchto dusíkatých látek je společně s dodávaným světlem a oxidem uhličitým spotřebovávána rostlinami, které vytvářejí kyslík, čímž se koloběh uzavírá. [1]

Vzhledem k biologické rovnováze a k potřebám živočichů a rostlin jsou v akvariijním systému důležité zejména následující parametry:

- teplota vody
- pH vody
- intenzita světla v čase

Dále je potřeba dodávat krmení, měnit vodu v akváriu, odkalovat ze dna přebytečné zbytky krmení a jiný odpad, případně přidávat hnojiva pro zlepšení růstu rostlin. Koloběh látek v akváriu je zobrazen na obrázku 1.



Obrázek 1: Zobrazení koloběhu látek v akváriu [1]

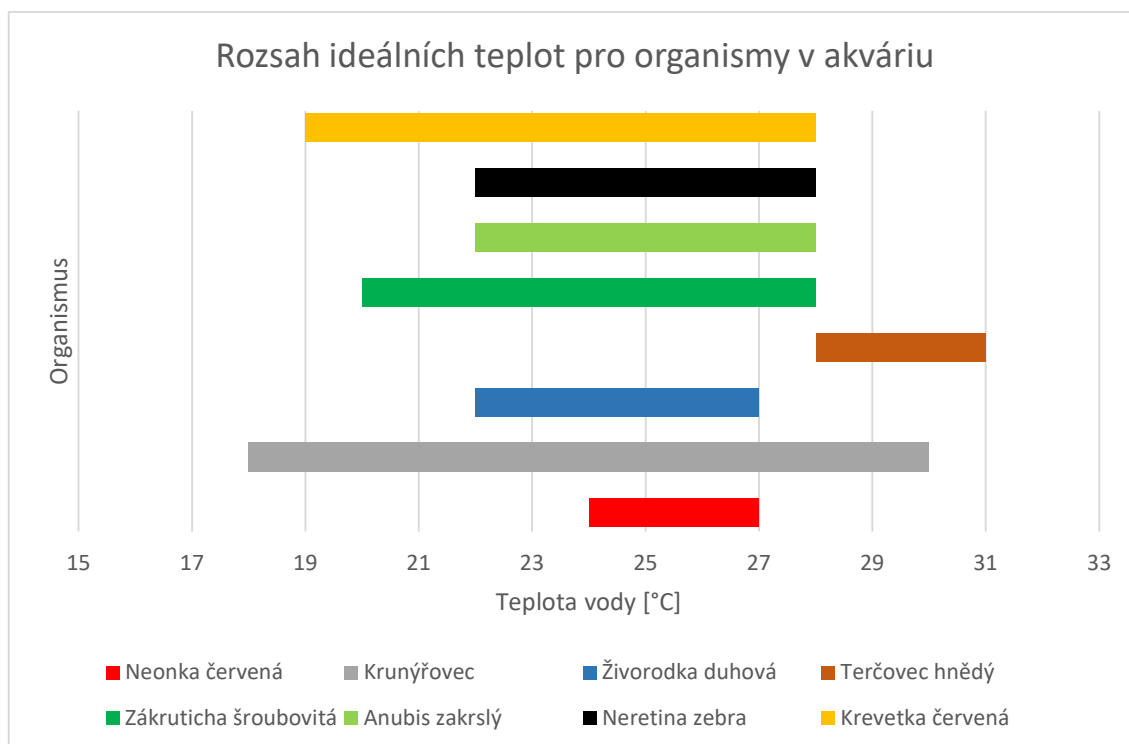
V akváriu, kde žijí živočichové a rostliny, se tvoří odpad především na bázi amoniaku. Ten je pak ve filtru přeměňován nitrifikačními bakteriemi na dusitany a dusičnany. Dále jsou dusičnany přeměňovány denitrifikačními bakteriemi na plynný dusík, který z vody uniká do atmosféry. Určitou část dusičnanů využijí i rostliny jako hnojivo.

1.1 Teplota vody

Teplota vody v akváriu je jedním z nejdůležitějších parametrů. U většiny sladkovodních tropických ryb, které jsou u nás chovány v drtivé většině případů, se vhodná teplota pro jejich chov obvykle pohybuje mezi 23 až 27 °C a u některých druhů to může být i více. Stručný přehled teplotních rozsahů pro vybrané druhy živočichů a rostlin (viz tabulka 1).

Tabulka 1: Přehled rozsahů teplot pro živočichy a rostliny v akváriu [2]

Druh	Organismus	Teplota [°C]
Neonka červená (<i>Paracheirodon axelrodi</i>)	Ryba	24 až 27
Krunýřovec sp. (<i>Ancistrus sp.</i>)	Ryba	18 až 30
Živorodka duhová (<i>Poecilia reticulata</i>)	Ryba	22 až 27
Terčovec hnědý (<i>Symphysodon aequifasciatus axelrodi</i>)	Ryba	28 až 31
Zákruticha šroubovitá (<i>Vallisneria spiralis</i>)	Rostlina	20 až 28
Anubis zakrslý (<i>Anubias barteri Schott var. nana</i>)	Rostlina	22 až 28
Neritina zebra (<i>Neritina natalensis</i>)	Plž	22 až 28
Krevetka červená (<i>Neocaridina davidi</i>)	Kreveta	19 až 28



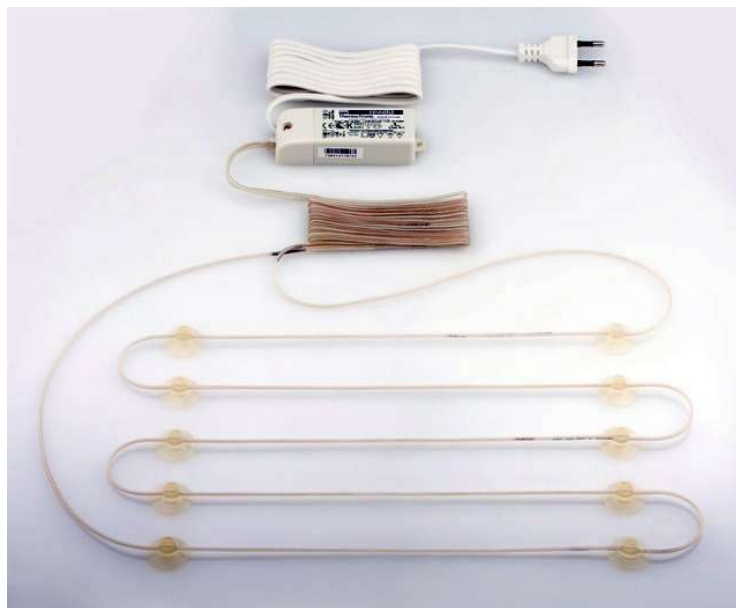
Obrázek 2: Zobrazení Tabulky 1: Přehled rozsahů teplot pro živočichy a rostliny v akváriu

Z výše uvedených údajů vyplývá, že nejnáročnějšími obyvateli akvárií na teplotu bývají ryby. Ty obvykle udávají požadavek na maximální rozsah teploty, a tedy na přesnost měřící a regulační techniky.

V akvaristice se prosazují dva typy ohřevu vody. Tím prvním a více používaným je klasické topné tělísko tvořené topnou spirálou zapouzdřenou do skleněného obalu, běžně opatřené termostatem. Výhodou je snadná instalace a jednoduché nastavení požadované teploty, které se provádí buď nastavením termostatu na požadovanou teplotu anebo je termostat rovnou z výroby pevně nastaven na jednu teplotu. Nevýhodou je fakt, že tento typ topného tělesa produkuje teplo pouze lokálně a nezajišťuje rovnoměrný ohřev vody v akváriu. Je tedy potřeba zvolit vhodnou polohu topného tělesa tak, aby teplo, které vyprodukuje, bylo rovnoměrně rozvedeno do celého akvária. To se obvykle řeší umístěním topného tělesa poblíž filtru. Ukázka komerčně vyráběného topného tělesa je na obrázku 3.



Obrázek 3: Topné těleso [3]

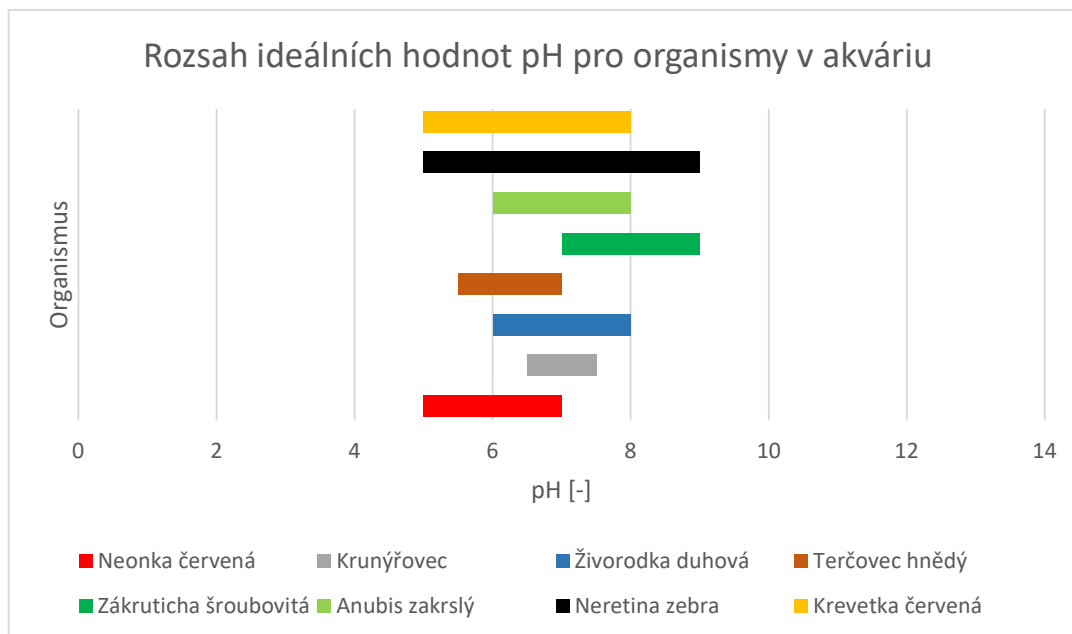


Obrázek 4: Topný kabel [4]

Druhým způsobem je vytápění pomocí topného kabelu, který je položen pod substrátem v akváriu. Lze si jej představit jako klasické podlahové topení. Nevýhodou je relativně složitá instalace a několikanásobná cena oproti topnému tělesu upevněném na stěně akvária. Výhodou je pak rovnoměrnější vytápění akvária a cirkulace vody v substrátu, což napomáhá šíření živin ze substrátu ke kořenům rostlin. Ukázka komerčně vyráběného topného kabelu viz obrázek 4.

1.2 pH vody

Hodnota pH vody v akváriu je další zásadní parametr pro akvaristy. pH je hodnota, které vyjadřuje, zda roztok reaguje kyselé nebo naopak zásaditě. pH je definováno jako záporně vzatý dekadický logaritmus aktivity oxoniových kationtů. Určuje se na logaritmické stupnici s rozsahem hodnot 0 až 14. Neutrální voda má pH 7. Každý druh má svoje ideální rozmezí pH, které se u většiny běžně chovaných ryb pohybuje v rozsahu jedné jednotky pH viz obrázek 5. Stejná situace je i u rostlin. [2]



Obrázek 5: Rozsah ideálních hodnot pH pro organismy v akváriu

Hodnota pH v akváriu během dne kolísá. Jedná se o přirozený jev, který se odehrává i v přírodě. Během dne, kdy rostliny fotosyntézou spotřebovávají oxid uhličitý, a tak vytvářejí kyslík, pH stoupá. V noci, kdy rostliny pouze dýchají a tím CO_2 produkují, pH klesá. V zarostlých akváriích se tak hodnota pH během dne změní i o jednu jednotku. Stejně jako u teploty, tak i u pH existují situace (například vykulení plůdků), kdy je potřeba jeho hodnotu držet konstantní.

Změna pH vody se provádí změnou množství oxidu uhličitého ve vodě. Pokud oxid uhličitý přidáme, část se změní v kyselinu uhličitou a pH klesne [5]. Pokud přestaneme oxid uhličitý do akvária dodávat, rostliny během dne zajistí růst hodnot pH. Případně je možné začít do vody v akváriu dodávat vzduch. Tím se z vody začne rychleji uvolňovat plyn CO_2 a pH mírně vzroste.

1.3 Intenzita osvětlení

Kvalitní osvětlení je nedílnou součástí každého akvária. Kvůli snížení rychlosti růstu řas se doporučuje akvária instalovat mimo přímý sluneční svit, a tak je umělé osvětlení nutností. Většina akvarijních ryb chovaných u nás pochází z tropických oblastí, které jsou typické přibližně stejně dlouhým dnem i nocí. Akvaristi, kteří se toto snaží napodobit, proto spouštějí osvětlení akvária na 10 až 12 hodin denně. Při snaze se co nejvíce přiblížit přírodním podmínkám, lze světlo na začátku dne pomalu rozsvěcovat, a tedy napodobovat východ slunce, obdobně tak večer se zhasínáním. Tento pomalý přechod je důležitý zejména pro ryby, které jsou rychlou změnou světla stresovány.

Vhodná intenzita osvětlení pro chov akvarijních ryb a pěstování rostlin udávána u dna akvária se pohybuje přibližně od 5 do 60 W/m^2 v závislosti na druhu žijícím v akváriu.

2 POUŽITÁ PLATFORMA, OPERAČNÍ SYSTÉM A VÝVOJOVÉ PROSTŘEDÍ

Jako řídicí jednotka v rámci této práce bude použita vývojová deska od firmy NXP Semiconductors, osazena čipem i.Mx8DV. Jako operační systém bude použit systém reálného času QNX Neutrino.

2.1 i.Mx8

i.Mx je rodina aplikačních procesorů od firmy NXP Semiconductors. Jedná se o SOC (System on chip). Verze i.Mx8DV není veřejně dostupnou, jedná se o interní verzi určenou k vývoji a testování pro účely firmy NXP Semiconductors.

2.1.1 System on chip

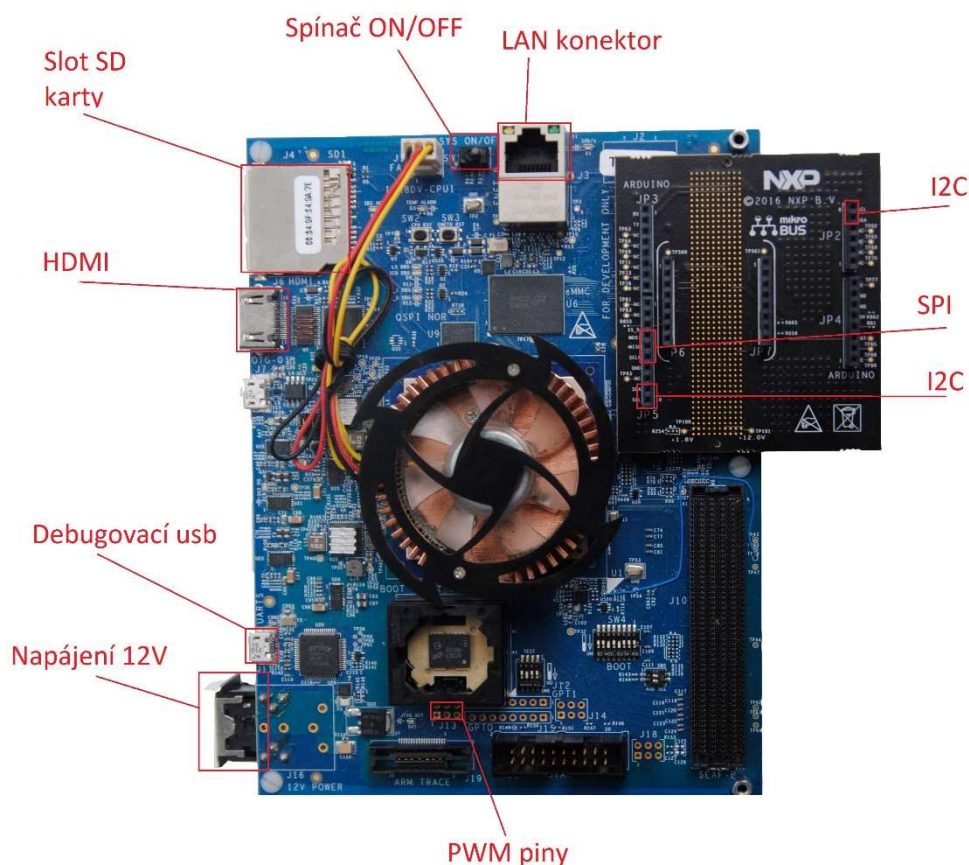
System on chip představuje architekturu integrující všechny součásti počítače nebo jiného elektronického zařízení do jednoho čipu. Tento čip většinou obsahuje také různé periferie, jako jsou například bloky pro zpracování analogových a digitálních signálů nebo pro práci s bezdrátovou technologií. Typickou aplikací je pak jeho využití ve vestavěných systémech. [15]

Architektura obsahuje tedy vždy vedle vlastního výkonového procesoru i cache potřebnou pro jeho práci. Užití tohoto systému se vyskytuje nejčastěji v malých, ale stále složitějších elektronických zařízeních. S architekturou SOC se setkáváme často v systémech jako jsou mobilní telefony, domácí elektronika, herní konzole apod. [15]

2.1.2 i.Mx8DV

i.Mx8DV je vývojová deska určená k testování a vývoji. Čip i.Mx8DV obsahuje procesorová jádra architektury ARM: 4x Cortex A53, 2x Cortex A72 a 2x Cortex M4. Pro potřeby této práce jsou důležité zejména periferie GPIO, I2C, PWM. Dále Ethernet a grafický výstup. Deska je napájena stejnosměrným napětím 12 V.

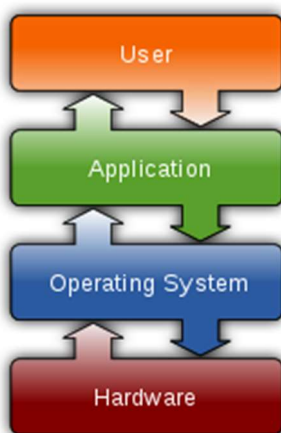
Čipy i.Mx8DV jsou určeny pro použití zejména v oblasti autoinfotainmentu, kde se starají například o chod palubního počítače, obsluhu GPS, přehrávání hudby či videa nebo připojení mobilního telefonu. K těmto účelům je zde využíváno audio/video rozhraní, dotyková obrazovka či tlačítka. i.Mx8DV však není určen pro ovládání systémů automobilu přímo souvisejících s řízením vozu.



Obrázek 6: Popis i.Mx8DV

2.2 Operační systém

Operační systém představuje základní programové vybavení počítače. Skládá se z jádra a pomocných systémových nástrojů. Hlavní úkoly operačního systému zahrnují umožnění uživateli ovládat počítač, vytvoření stabilního aplikačního rozhraní a přidělování systémových zdrojů pro procesy. [8]



Obrázek 7: Zobrazení vztahu mezi operačním systémem, HW, aplikačním SW a uživatelem[12]

Nad každým druhem systémů je třeba uvažovat samostatně vzhledem k použitému operačnímu systému – například tam, kde procesor bude pouze číst data z A/D převodníku a po sběrnici je bude posílat dál, nemá význam zavádět jakýkoliv operační systém. [6]

Pokud není použit operační systém, program je složen pouze z ovladačů hardwaru a vlastního programu. Tento přístup má nejmenší nároky na výkon procesoru a je možné ho při pečlivém zvážení použít i v časově-kritických aplikacích. [7]

2.2.1 Operační systém reálného času

Operační systém reálného času (real-time operating system, RTOS) je typ operačního systému, který umožňuje reagovat na události v okolí počítače průběžně, tedy v reálném čase (s naprosto zanedbatelným zpožděním). RTOS poskytuje záruky, že činnost bude v určitém časovém úseku dokončena. Pro RTOS je charakteristické, že správná funkce nezávisí jen na početním výsledku, ale i na čase, kdy ho bude dosaženo. Běžné RTOS fungují tak, že časové požadavky na výpočet typicky nezaručuje naprosto dokonale – negarantuje je, ale snaží se k takovým výsledkům co nejvíce přiblížit.

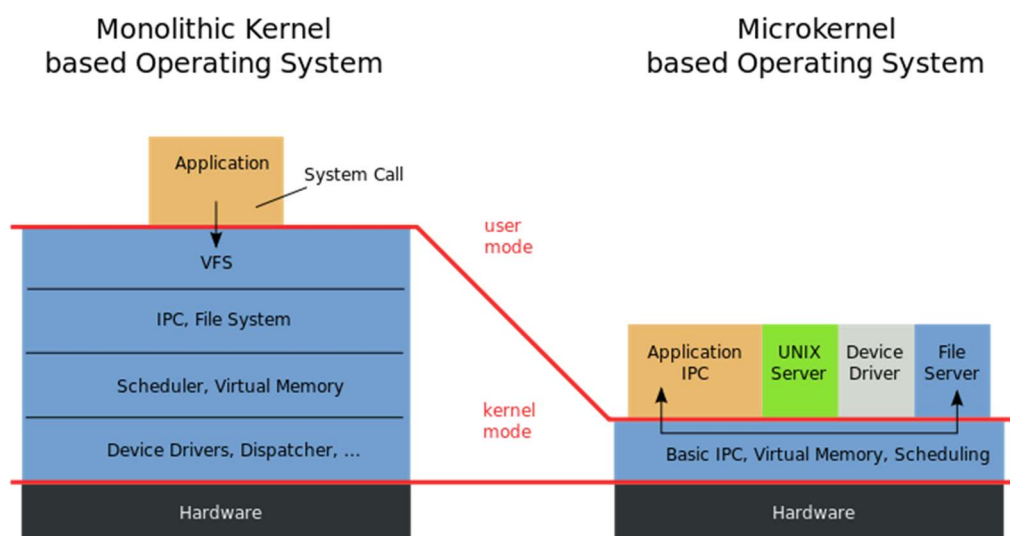
Podle toho dělíme RTOS na dva podtypy:

- HARD RTOS – úloha se musí stihnout do stanoveného časového limitu. Nestihnutí může mít katastrofální následky. Příkladem může být ABS v autě. [16]
- SOFT RTOS – úloha by se měla stihnout do časového limitu, dovolují se drobné odchylky v časových reakcích. Například jedna úloha z tisíce se nestihne v daném časovém limitu. [16]

2.2.2 Monolitické jádro a mikrojádru

Mikrojádru je velmi malé a obsahuje jen ty nejzákladnější funkce. Typicky se jedná o správu paměti, plánování procesů a meziprocessorovou komunikaci. Tím se minimalizuje objem kódu běžící v privilegovaném režimu. Ostatní potřebné části jádra jsou řešeny v uživatelském prostoru jako běžné procesy. Výhodou této architektury je možnost jednotlivé nepotřebné komponenty vypnout, případně zase zapnout. S tím souvisí další vlastnost mikrojádrového operačního systému, a to, že v případě poruchy libovolné komponenty systém neseleže a komponentu restartuje. Nevýhodou mikrojádru oproti monolitickému je nižší výkon. [9]

Monolitické jádro je druh jádra operačního systému, jehož veškerý obsah běží ve stejném paměťovém prostoru. Chyba v jednom subsystému tak může zablokovat jiný, nebo dokonce zapříčinit pád celého jádra. [10]



Obrázek 8: Rozdíl mezi monolitickým jádrem a mikrojádrem [11]

2.2.3 QNX Neutrino

Jedná se o komerční operační systém reálného času, který se většinou používá v embedded (vestavěných) systémech. Výrobce BlackBerry poskytuje několik variant tohoto systému, které splňují nejrůznější certifikace. Díky tomu je systém používán nejen v průmyslu, obraně a medicíně, ale i jiných kritických aplikacích, kde je kladen důraz na velkou stabilitu systému. [7]

Na rozdíl od Linuxu, který používá monolitické jádro, je QNX postaven na architektuře mikrojádra. QNX Neutrino je kompatibilní se standardem POSIX, což umožňuje snadné portování aplikací napsaných pro jiný Unix-like systém. [7]

QNX souborové systémy jsou obsluhovány mimo mikrojádro v uživatelské chráněné paměti. Vývojáři mohou spustit, zastavit nebo aktualizovat souborové systémy za běhu bez nutnosti restartu. Více souborových systémů může být spuštěno současně. [13]

2.3 Framework Qt

Pro vytvoření grafického uživatelského rozhraní v této práci je použit framework Qt. Jedná se o aplikační rámec (framework) široce používaný pro vytváření aplikačního softwaru s grafickým uživatelským rozhraním pro rozličné softwarové a hardwarové platformy. [30] Qt je napsaný v jazyce C++. V Qt je jazyk C++ mírně rozšířen a jsou mu přidána některá nová klíčová slova jako *signals*, *slots* a další. [31]

Qt framework poskytuje základní stavební bloky (různé typy funkcí, tříd a datových struktur), ze kterých je možné stavět aplikaci. Obsahuje širokou škálu přizpůsobitelných widgetů, grafických pláten, stylů atd., které lze využít k tvorbě grafických uživatelských rozhraní. V Qt je také obsažena podpora pro Internet, síť, databáze, multimédia, zvuk,

video, grafiku, vizuální efekty a animace. Kromě samotných knihoven Qt nabízí řadu multiplatformních vývojových nástrojů. V první řadě je to vývojové prostředí Qt Creator. [31]

2.3.1 Licence Qt

Qt framework je dostupný pod celou řadou typů licencí, takže v něm lze dle libosti (samozřejmě za předpokladu dodržení daného typu licence) vyvíjet komerční, nekomerční, otevřené nebo i uzavřené aplikace, aniž by se vývojář vystavoval riziku jakéhokoliv postihu. [31]

Svobodné licence Qt - pro nekomerční vývoj a studium je Qt nabízeno pod svobodnými licencemi. Můžeme volit mezi GNU LGPL (Lesser General Public License) v. 2.1 nebo GNU GPL (General Public License) v. 3.0. [31]

Licence LGPL nabízí možnost vyvíjet (otevřené) nekomerční, ale i proprietární (komerční) aplikace, avšak musí být splněny podmínky této licence. V praxi to znamená, že vývojář může vytvořit aplikaci, kterou může i prodávat, aniž by musel zveřejnit její zdrojový kód. Musí ovšem zveřejnit zdrojový kód, který se týká úprav vlastního Qt frameworku. [31]

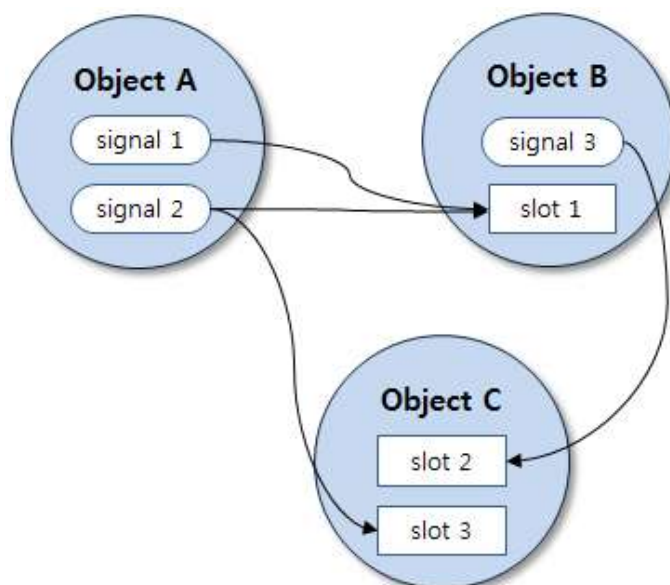
V případě striktnější licence GPL je pak vývojář omezen více. U aplikací vytvořených pod touto licencí, musí být zveřejněn jejich veškerý zdrojový kód a musí být splněny ostatní podmínky této licence. [31]

Komerční licence Qt - když se vývojář rozhodne vyvíjet komerční aplikace, u kterých bude potřebovat pozměnit nebo jinak upravit vlastní Qt framework, či využít statické sestavní aplikace s Qt frameworkem a nebude chtít zveřejňovat zdrojové kódy takových aplikací, musí zvolit jedinečnou *komerční licenci Qt (Qt Commercial Developer Licence)*. [31]

2.3.2 Signály a sloty

Signály a sloty jsou asi největší předností Qt. Je to způsob komunikace mezi jednotlivými objekty. Signály jsou jednotlivými objekty vysílány. Objekt, který jakýmkoliv způsobem změnil svůj stav, vyšle signál a nestará se o to, jestli byl jiným objektem přijat nebo ne. Objekt, který přijme konkrétní signál spustí předem definovaný slot. Sloty slouží k reakci na určitý signál. Aby mohl objekt na daný signál reagovat, je nutné nejprve v tomto objektu spojit konkrétní signál s konkrétním slotem.

Signály a sloty se spojují pomocí funkce *connect*. Případně je možné signály a sloty daných objektů propojit v grafickém prostředí programu Qt Creator. To se provede kliknutím pravým tlačítkem myši na objekt. Ze zobrazené nabídky se zvolí možnost *Go to slot* a vybere se požadovaný signál. Program připraví kostru slotu a automaticky vytvoří spojení mezi signálem a slotem. Vývojáři pak stačí dopsat implementaci daného slotu.



Obrázek 9: Znáznornění signálů a slotů [33]

2.3.3 Základní třídy Qt

Qt je postaveno na objektovém modelu, a to mimo jiné znamená, že jeho jednotlivé třídy tvoří jistou hierarchii, na jejímž vrcholu se nachází právě tyto základní třídy. [31]

Třída QApplication. Tato třída tvoří jádro všech konzolových aplikací napsaných v Qt. Inicializuje a ukončuje aplikaci, poskytuje jí hlavní smyčku událostí a také zpracovává a rozesílá události od operačního systému a dalších zdrojů objektům aplikace. [31]

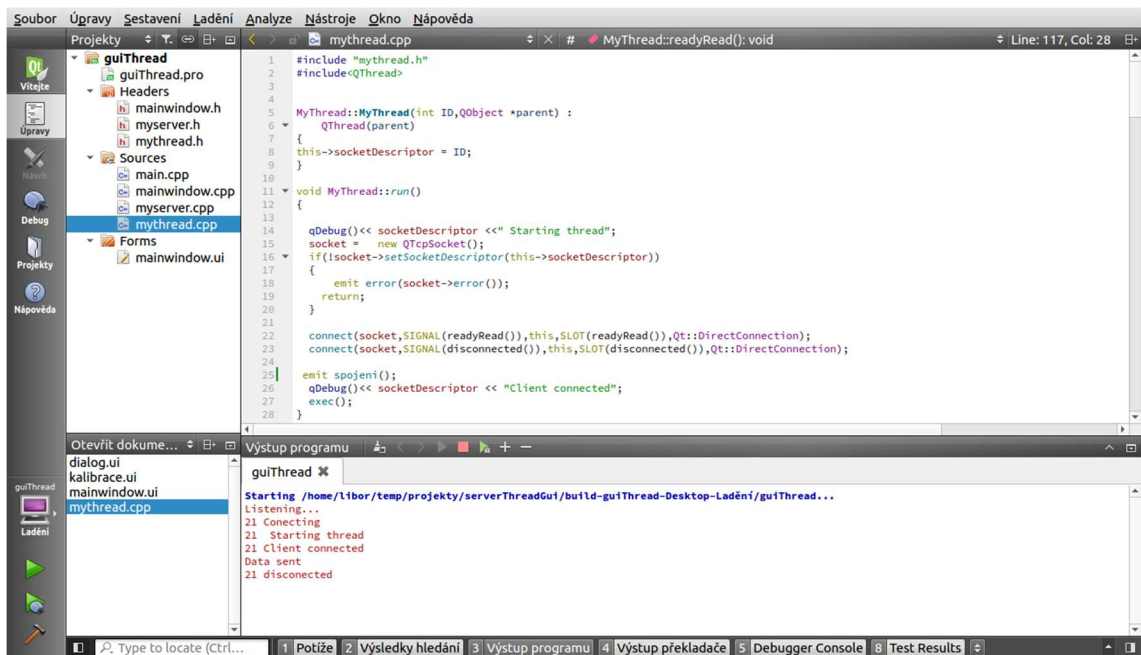
Třída QApplication je odvozena z třídy QApplication a přebírá všechny její vlastnosti a funkce. Navíc přidává své vlastní pro podporu GUI aplikací. Její objekt, který by měl být v programu vytvořený jako první, je hlavním objektem GUI aplikace a stará se o celý její chod od inicializace po ukončení. [31]

Třída QObject je базovou třídou pro téměř všechny Qt třídy. Zajišťuje mechanismus signálu a slotu a poskytuje mu všechny potřebné funkce a atributy. Zapouzdřuje funkci connect. Do kompetence této třídy patří také správa názvu objektu a dalších metainformací sloužících k identifikaci typu objektu. Dále třída umožňuje podporu událostí a jejich filtrů, vláken a nízkoúrovňových časovačů. [31]

Třída QWidget je базovou třídou všech tříd prvků uživatelského rozhraní, tzv. widgetů. Widget je základním stavebním prvkem uživatelského rozhraní, který přijímá události od okenního systému a dalších zdrojů a vykresluje svoji vizuální reprezentaci na obrazovku. Jakýkoli widget může tvořit vlastní okno. [31]

2.4 Vývojové prostředí Qt Creator

Qt Creator je IDE (integrated development enviroment), které je součástí SDK (Software development kit) pro Qt framework. Toto vývojové prostředí je určené pro tvorbu grafických aplikací, které jsou nezávislé platformě. Je vytvořené speciálně pro vývojáře, kteří využívají framework Qt. Součástí Qt Creatoru je Qt Designer, který slouží k jednoduchému grafickému navrhování formulářů a v omezené míře umožňuje také spojování signálů a slotů. Dále v sobě integruje Qt Asisstant, který slouží k jednoduchému přístupu k dokumentaci Qt přímo z IDE [32]. Qt Creator zvýrazňuje syntaxi a zobrazuje metody daného objektu.



Obrázek 10: Zobrazení Qt Creatoru

3 NÁVRH HW ROZHRAŇÍ

Cílem této kapitoly je navrhnout vhodné rozhraní, které bude sloužit k propojení čidel a akčních členů s řídící deskou. Návrh desky vychází z požadavků, které vyplývají z teoretického rozboru v této práci (1). Jelikož pH bude měřeno analogově (bude vypočteno z napětí), jeví se jako vhodná varianta měřit i ostatní veličiny analogově. Protože na vývojové desce nejsou dostupné analogové vstupy, bude potřeba tyto hodnoty převést do digitální podoby a poté přes vybranou sběrnici poslat na řídící desku.

3.1 Teplotní čidlo

Potřebná přesnost měření teploty byla v návaznosti na teoretický rozbor (1.1) zvolena na $\pm 0,5 \text{ }^{\circ}\text{C}$. V principu se při elektrickém měření teploty využívá materiál, který při změně teploty mění své elektrické vlastnosti. Většinou se jedná o změnu elektrického odporu v závislosti na teplotě. Takové součástky nazýváme termistory. Pokud se změní elektrický odpor a prvkem protéká konstantní elektrický proud, pak se podle Ohmova zákona projeví tato změna i změnou napětí na daném prvku. V praxi se pak využívá zapojení takového prvku do série s rezistorem s pevně daným elektrickým odporem, aby bylo možné měřit změnu napětí na daném prvku.

Kromě samotné přesnosti daného teplotního čidla je důležitá také jeho reakční doba. Ta by měla být přiměřená rychlostem změny teploty. Vzhledem k velké tepelné kapacitě vody a oproti tomu malému výkonu běžně používaných topných těles se teplota v akváriu mění velmi pomalu (jednotky minut až jednotky hodin v závislosti na objemu akvária a výkonu topného tělesa). Z toho důvodu nejsou na reakční dobu čidla kladeny žádné zvláštní požadavky.

3.1.1 Termočlánek

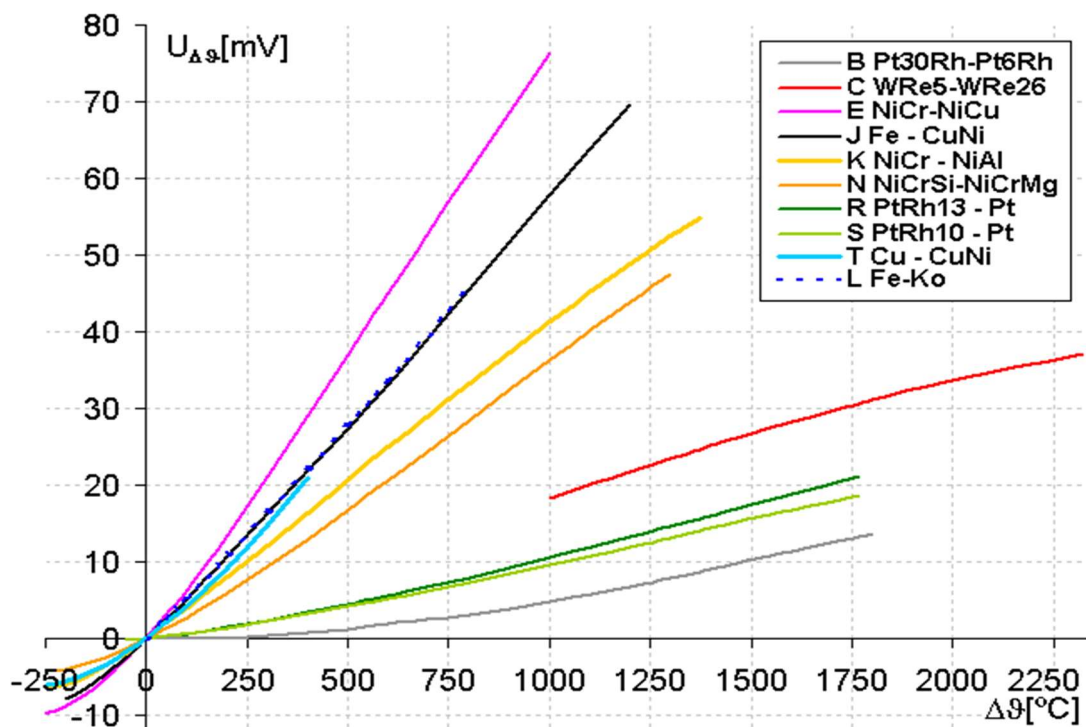
K měření teploty lze využít tzv. termočlánek. Jedná se o spojení dvou různých kovů do série. Na základě termoelektrického jevu lze na koncích těchto vodičů měřit napětí, které je úměrné teplotě spoje těchto kovů.

Termočláanky se používají většinou pro měření vysokých teplot např. v pecích. Termočlánek není vhodný pro přímé měření přesných hodnot teploty kvůli své malé citlivosti. Například termočlánek typu E změní své termoelektrické napětí o 10 mV při změně teploty o 125 $^{\circ}\text{C}$ viz obrázek 11. Na obrázku jsou vidět charakteristiky běžně používaných termočláanků.

Takto malá citlivost odpovídá změně 40 μV při změně 0,5 $^{\circ}\text{C}$. To klade velké nároky na AD převodník, který bude daný signál zpracovávat. Velikost kvantizačního kroku by tedy musela být menší než 40 μV . Při takto malé citlivosti bude obvod také velmi náchylný na případné rušení, které by se mohlo na signálové cestě objevit.

Výpočet velikosti kvantizačního kroku q:

$$q = \frac{U_{ref}}{2^N} [V] \quad (1)$$



Obrázek 11: charakteristiky termočlánků [19]

Kde U_{ref} je referenční napětí převodníku, 2^N je počet kvantizačních hladin a N je rozlišení převodníku.

Po úpravě dostáváme:

$$N = \log_2\left(\frac{U_{ref}}{q}\right) [-] \quad (2)$$

Po dosazení za U_{ref} a q :

$$N = \log_2\left(\frac{5}{40 \cdot 10^{-6}}\right) = 16,93 [-] \quad (3)$$

Při použití termočlánku je potřeba k dosažení dostatečné přesnosti použít nejméně 17 bitový převodník.

3.1.2 Termistor

Pokud hodnota elektrického odporu roste spolu s teplotou, jedná se o termistor PTC (Positive Temperature Coefficient), tím je například čidlo PT100. Klesá-li hodnota elektrického odporu s rostoucí teplotou, jedná se o termistor NTC (Negative Temperature Coefficient).

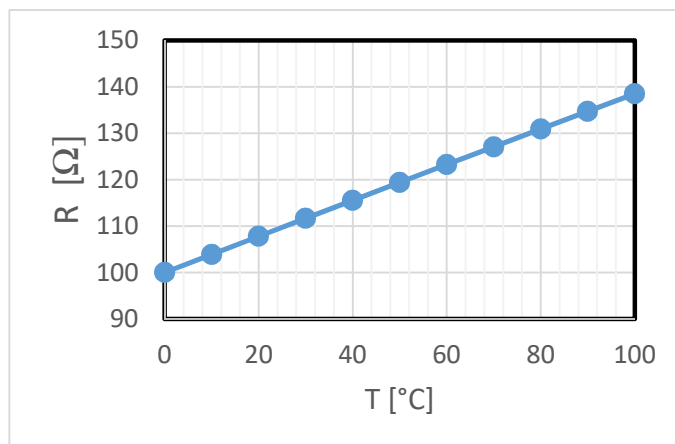
Pro určení nejmenšího možného rozlišení AD převodníku je nutné nejprve určit citlivost čidla. Pro uvažované zapojení (viz příloha A.1) se vstupní napětí na převodníku AD určí jako výstupní napětí odporového děliče podle rovnice (4)

$$U_{out} = \frac{U_{ref}}{R_1 + R_{temp}} \cdot R_{temp} [V] \quad (4)$$

U_{ref} je napětí na děliči (v tomto případě 5 V), R_1 (kvůli proudové ochraně čidla) 5 k Ω a R_{temp} je hodnota odporu čidla PT100.

Tabulka 2: Teplotní závislost PT100 [17]

T [°C]	R [Ω]
0	100,00
10	103,90
20	107,79
30	111,67
40	115,54
50	119,40
60	123,24
70	127,08
80	130,90
90	134,71
100	138,51



Obrázek 12: Závislost odporu PT100 na teplotě [17]

Při teplotě 20 °C $R_{temp} = 107,79 \Omega$:

$$U_{out} = \frac{5,00}{500,00 + 107,79} \cdot 107,79 = 0,1055 \text{ V} \quad (5)$$

Při teplotě 30 °C $R_{temp} = 111,67 \Omega$

$$U_{out} = \frac{5,00}{500,00 + 111,67} \cdot 111,67 = 0,1092 \text{ V} \quad (6)$$

Na základě výpočtu (5) a (6) lze určit citlivost snímače jako

$$k = \frac{\Delta U_{out}}{\Delta T} = \frac{0,1092 - 0,1055}{30 - 20} = 0,37 \text{ mV/}^\circ\text{C} \quad (7)$$

kde k je citlivost, ΔU_{out} je rozdíl napětí při teplotě 30 a 20 °C, ΔT je potom rozdíl teploty, v tomto případě 10 °C.

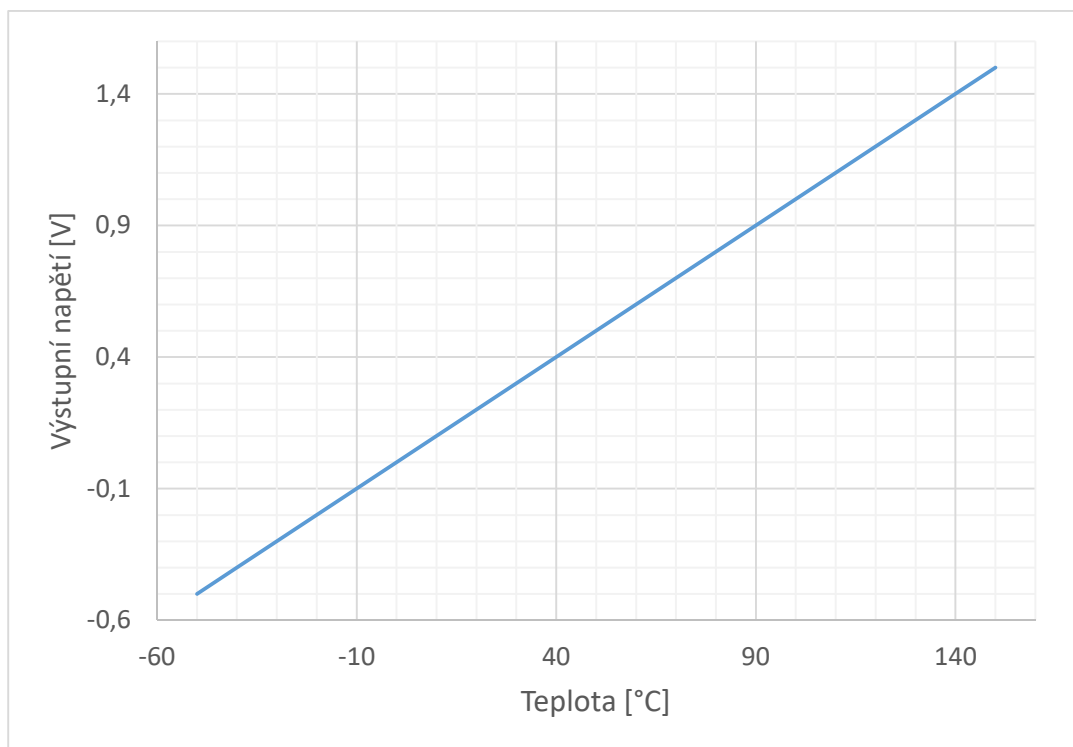
Určená citlivost k je 0,37 mV na 1 °C, tedy 0,185 mV/0,5 °C. Potřebná délka bitového slova převodníku při použití čidla PT100:

$$N = \log_2 \left(\frac{U_{ref}}{q} \right) = \log_2 \left(\frac{5}{0,185 \cdot 10^{-3}} \right) = 14,72 \quad (8)$$

Při použití čidla PT100 je tedy nutné použít AD převodník s minimální délkou bitového slova 15 bitů.

3.1.3 Integrovaný obvod LM35

Teplotu lze měřit i pomocí integrovaných obvodů. Jedním z nich je obvod LM35. Jedná se o převodník teploty na napětí. Převodní koeficient tohoto obvodu je 10 mV/°C. Stejnosměrné napájecí napětí je v rozmezí 4 až 30 V. Obvod vykazuje malé samozahřívání 0,08 °C ve vzduchu. Přesnost měření je výrobcem udávána 0,5 °C. [21]



Obrázek 13: Závislost výstupního napětí na teplotě čidla LM35 [20]

Potřebná délka bitového slova převodníku při použití obvodu LM35:

$$N = \log_2(\frac{U_{ref}}{q}) \quad [-] \quad (9)$$

Po dosazení U_{ref} a q :

$$N = \log_2(\frac{5}{10 \cdot 10^{-3}}) = 8,97 [-] \quad (10)$$

Při použití obvodu LM35 stačí použít 9 bitový převodník. Pokud bude použit 12 bitový převodník bude rozlišení $0,125^{\circ}\text{C}$.

Vzhledem k dobré dostupnosti, ceně, přesnosti a nárokům na následné zpracování signálu byl zvolen obvod LM35 ve vodotěsném zapouzdření.

3.2 Topné těleso

Pro ohřev vody v akváriu lze použít libovolné akvarijní topné těleso s dostatečným výkonem vzhledem k objemu akvária. Toto těleso bude připojeno do zásuvky se síťovým napětím 230 V, která bude řízená aplikací na základě teploty v akváriu. Jako ochrana proti přehřátí vody při poruše systému slouží nezávislý termostat na topném tělese, který bude nastaven o 1°C víc než požadovaná teplota. Aby se v případě poruchy zabránilo vychladnutí vody bude v akváriu umístěno druhé topné těleso s vlastním termostatem nastaveným o 1°C pod požadovanou teplotu.

3.3 pH senzor

Z již uvedených rozdílných nároků ryb a rostlin na požadované pH vyplynula přesnost měření $\pm 0,5$ stupně pH. (1.2)

Způsobů měření pH kapalin existuje více. Například je to použití acidobazického indikátoru, který reaguje při kontaktu s měřenou kapalinou změnou barvy. Výsledná barva je porovnávána s ocejchovanou paletou barev, a tak je určeno pH. Tato metoda jde použít pro ruční měření pH v akváriu, ale pro automatizovaný proces je prakticky nepoužitelná.

Další metoda se nazývá potenciometrie. Tato metoda je založena na velmi přesném měření potenciálu mezi měrnou a referentní elektrodou. Odvozením z Nerstovy rovnice [26] teoreticky platí, že změna pH o jeden řád (např. z pH 7 na pH 8) vyvolá změnu napětí o 59 mV. V praxi se však tato hodnota ještě kalibruje pro každou sondu individuálně na základě vložení sondy do pufru [17]. Voltmetr pro takové měření musí mít velký vstupní odpor (minimálně $10^{14} \Omega$, kvalitní přístroje mají i o řád až dva větší).

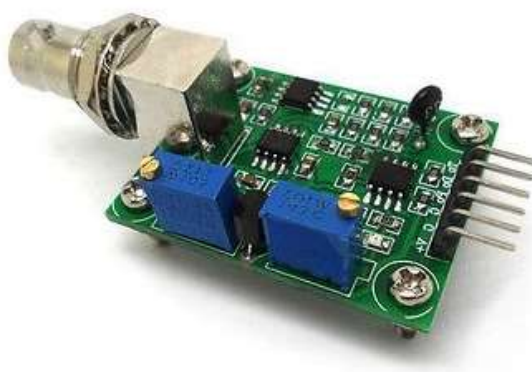


Obrázek 14: Ukázka profesionálního pH metr PH 100 ATC [28]

Protože návrh a výroba převodníku pH na napětí je mimo rámec této práce, je pro tyto účely použit komerčně dostupný modul pro převod napětí z pH sondy[27]. Výstupní napětí vybraného modulu má hodnotu 2,5 V pro pH 7. Koeficient změny napětí na změně pH je 0,11 V / stupeň pH. Při použití vzorce (2) a dosazení dostáváme

$$N = \log_2\left(\frac{5}{55 \cdot 10^{-3}}\right) = 6,51 [-] \quad (11)$$

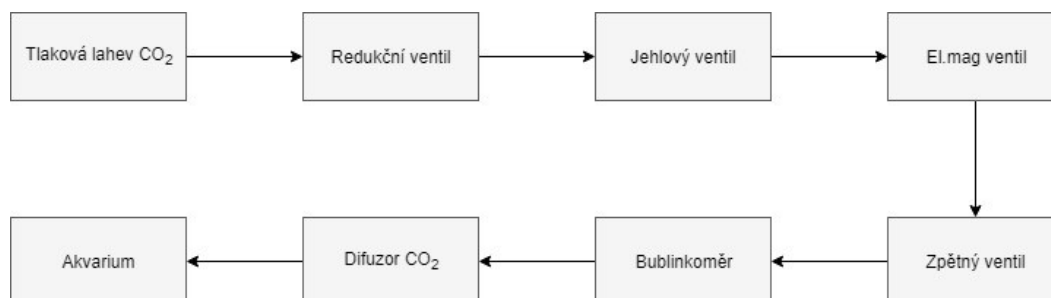
Pro dosažení potřebné přesnosti je nutné použít minimálně 7 bitový AD převodník.



Obrázek 15: Modul pro měření pH [14]

3.4 Dávkování CO₂

Oxid uhličitý se bude do akvária dodávat z tlakové lahve. Jedná se o jediné efektivní řešení. Samotná regulace pak spočívá v ovládání elektromagnetického ventilu, přes který proudí CO₂ do akvária. Elektromagnetický ventil je spínaný systém správy akvária na základě změřeného pH v akváriu. Celý řetězec pro dodání CO₂ do akvária je popsán na obrázku 16.



Obrázek 16: Řetězec dodání CO₂

Regulační ventil, jak název napovídá, slouží k regulaci tlaku za tlakovou lahví. Je zde potřeba z důvodu velkého tlaku uvnitř lahve. Tlak v tlakových lahvích s CO₂ je od 2 do 7 MPa. Běžně dostupné elektromagnetické ventily jsou konstruovány na maximální tlak 1 MPa. Jehlicový ventil se používá pro jemné doladění tlaku. Elektromagnetický ventil bude ovládán řídicí jednotkou a slouží k regulaci průtoku CO₂ do akvária. Zpětný ventil se do řetězce řadí proto, aby nemohla voda z akvária protékat směrem k tlakové lahvi, pokud je lahev pod úrovní akvária. Bublinkoměr se do řetězce řadí pro snadné měření průtoku CO₂. Úlohou difuzéru je rozptýlit plyn na co nejmenší bublinky tak, aby se co nejlépe rozpouštěly ve vodě.

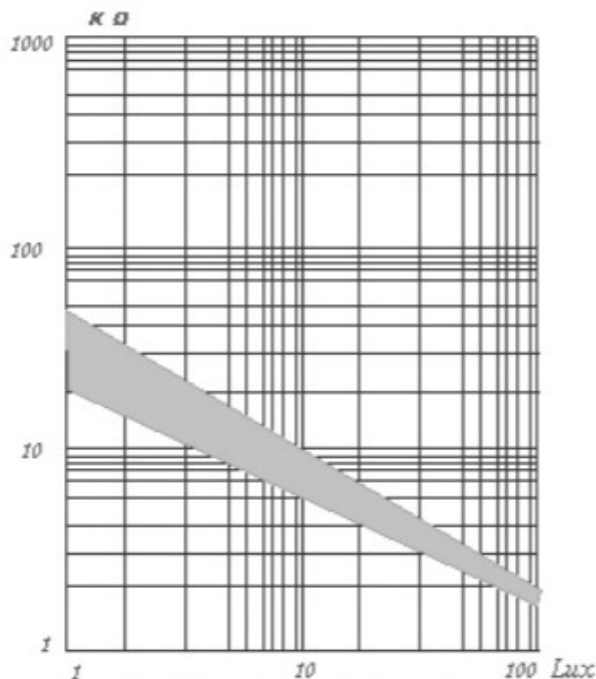


Obrázek 17: Zobrazení tlakové lahve s ventilem [29]

3.5 Senzor intenzity světla

Obdobně jako se u měření teploty využívají prvky měnící své vlastnosti při změně teploty, u měření intenzity osvětlení se používají prvky, které mění své elektrické vlastnosti při změně intenzity dopadajícího světla. Jedná se o fotorezistor, fototranzistor a fotodiodu. Princip je u těchto součástek podobný. U fotorezistoru dopadající foton na základě fotoelektrického jevu předá energii elektronu. Tím se změní vodivost daného materiálu (například sulfid kadmia). U fototranzistoru dopadající foton na emitorový PN přechod způsobuje větší otevření tranzistoru. U fotodiody je osvětlen přechod PN, kde je na základě dopadajících fotonů zvětšena vodivost tohoto přechodu. Fotodioda dokáže reagovat z výše zmíněných součástek na změnu osvětlení nejrychleji.

Pro účel této práce by se dala využít libovolná výše zmíněná součástka. Na základě ceny, dostupnosti a jednoduchosti zapojení byl vybrán fotorezistor GL5516. Tento fotorezistor vykazuje odpor 500 k Ω při úplné tmě a 5-10 k Ω při intenzitě dopadajícího světla 10 Lux (viz obrázek 18).



Obrázek 18: Lux-Ohmová charakteristika fotorezistoru GL5516 [18]

Vstupní napětí na převodníku AD se určí jako výstupní napětí odporového děliče podle rovnice (12). Výsledné vstupní napětí bude v rozsahu 0,41 až 4,90 V viz výpočet (13) a (14).

$$U_{out} = \frac{U_{ref}}{R_1 + R_{foto}} \cdot R_{foto} \text{ [V]} \quad (12)$$

Kde U_{ref} je napětí na děliči (v tomto případě 5 V), R_1 je 10k Ω a R_{foto} je hodnota odporu fotorezistoru.

Při úplné tmě $R_{foto} = 500 \text{ k}\Omega$:

$$U_{out} = \frac{5}{10k + 500k} \cdot 500k = 4,902 \text{ V} \quad (13)$$

Při osvětlení 100 Lux $R_{foto} = 0,900 \text{ k}\Omega$

$$U_{out} = \frac{5}{10k + 0,9k} \cdot 0,9k = 0,413 \text{ V} \quad (14)$$

Na základě výpočtu (13) a (14) lze určit citlivost snímače jako

$$k = \frac{\Delta U_{out}}{\Delta Lux} = \frac{4,902 - 0,413}{100 - 0} = 44,9 \text{ mV/lux} \quad (15)$$

kde k je citlivost, ΔU_{out} je rozdíl napětí při osvětlení 100 a 0 lux, Δlux je potom rozdíl intenzit osvětlení, v tomto případě 100 a 0 lux.

Pokud by byla požadovaná přesnost měření intenzity stanovena na jeden lux, pak při použití vzorce (2) a dosazení dostáváme

$$N = \log_2\left(\frac{5}{44,9 \cdot 10^{-3}}\right) = 6,80 [-] \quad (16)$$

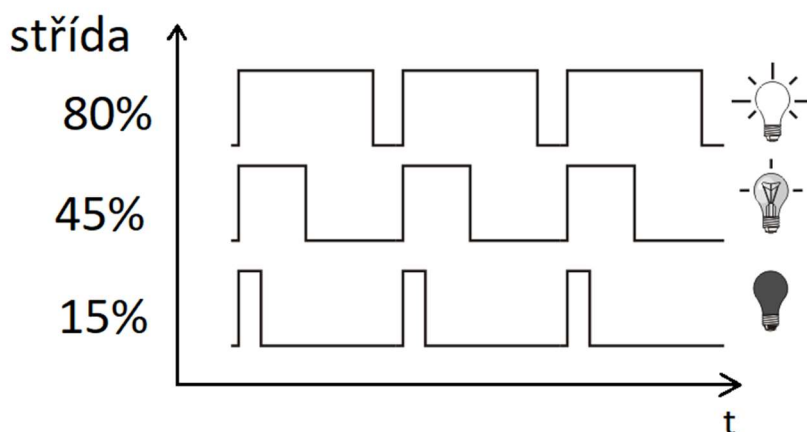
tedy potřebná délka bitového slova AD převodníku je minimálně 7 bitů.

3.6 Osvětlení

Vzhledem k potřebě postupně rozsvěcet a zhasínat osvětlení připadá v úvahu žárovka se žhaveným vláknem nebo LED osvětlení. LED osvětlení je výhodnější zejména výrazně lepší účinností a s tím spojeným sáláním tepla, které by bylo zejména v letních měsících nežádoucí. Standardní napětí pro LED svítidla je 12 V. Proto bude pro osvětlení určena výstupní svorka 12 V s PWM modulací pro plynulou regulaci intenzity světla.

3.6.1 PWM modulace

PWM (pulzně šířková modulace) je založena na změně šířky napájecích pulzů. V závislosti na šířce pulzu se přenese jen určité procento maximálního výkonu viz obrázek 19.



Obrázek 19: Zobrazení průběhu PWM [38]

3.7 A/D převodník

A/D převodník je elektronický obvod, který slouží k převodu analogového signálu

(zejména elektrického napětí) na digitální. Tento převod umožňuje číslicové zpracování původně analogového signálu.

3.7.1 Pojmy spojené s A/D převodníkem

Vzorkování je děj, při němž se v pravidelných časových intervalech zjišťuje okamžitá hodnota vzorkovaného signálu. Počet vzorků za sekundu určuje vzorkovací kmitočet. Příliš vysoký vzorkovací kmitočet znamená velké množství vzorků a jsou tedy kladeny větší nároky na paměť pro uchování vzorků a na rychlost převodníku. Příliš malý vzorkovací kmitočet zapříčiní zkreslení převáděného signálu. V takovém případě nastává chyba vzorkováním.

Pro správnou volbu vzorkovacího kmitočtu se využívá Shannon-Kotelnikovova teorému, podle něhož musí být vzorkovací kmitočet alespoň dvakrát vyšší než nejvyšší harmonická složka obsažená v převáděném signálu. Vzhledem k tomu, že v této práci se bude převádět pouze stejnosměrný signál nemá volba vzorkovacího kmitočtu vliv na zpracování signálu.

Kvantování je proces, který aktuální vstupní hodnotě přidělí právě jednu kvantizační hladinu. Počet bitů výstupního slova převodníku ukazuje na citlivost převodníku, neboť pro N-bitový převodník existuje právě 2^N kvantizačních hladin (rolišovacích hodnot). Rozdíl dvou kvantizačních hladin se nazývá kvantovací krok. Rozdíl mezi skutečnou hodnotou vstupního signálu a přidělenou kvantizační hladinou se nazývá chyba kvantování. Chyba kvantování vyjádřená v absolutní hodnotě může nabývat maximálně jedné poloviny kvantovacího kroku.

Doba přechodu vyjadřuje dobu trvání potřebnou k vystavení binárního kódu na výstup od doby, kdy je přiveden signál na vstup převodníku [23]. Vzhledem k charakteru měřených veličin není doba přechodu zásadní parametr pro volbu AD převodníku.

3.7.2 Základní typy A/D převodníků podle způsobu převodu

Komparační paralelní – jedná se o převodník s přímým typem převodu. Referenční napětí se rozdělí na odporovém děliči složeném z $N+2$ odporů, kde N je bitové rozlišení převodníku. Napěťové komparátory porovnávají převáděné napětí s referenčním, které jim vytvořil napěťový dělič. Převodník kódu vyhodnocuje výstupy z komparátorů a vytváří výstupní datové slovo. [22]

Kompenzační s postupnou aproximací – Aproximační registr odhaduje velikost vstupního signálu a generuje výstupní číslicový kód. Na aproximační registr je připojen DA převodník, který zajišťuje převod odhadovaného slova na analogovou veličinu. Vstupní a odhadované napětí porovnává napěťový komparátor. [22] Aproximační registr postupně pomocí metody půlení intervalů upravuje výstupní datové slovo. Aproximace skončí, když je rozdíl mezi vstupním a odhadovaným napětím menší než kvantizační krok. Výhodou tohoto převodníku je jednoduchá konstrukce, nevýhodou je proměnlivá doba převodu.

Sigma-delta převodník je A/D převodník využívající sigma-delta modulaci. Tato modulace má rozlišení 1 bit. Výstupní signál má tedy šířku bitového slova 1 bit. Z toho vyplývá, že informace o napěťové úrovni nemůže být uchována v amplitudě (úrovni) výstupního signálu. Namísto toho je tato informace nesena hustotou jedniček v čase. Čím

vyšší je úroveň vstupního analogového signálu, tím vyšší „hustota jedniček“ je v digitálním signálu. [24] Výhodou je velké celkové rozlišení převodníku využívající sigma-delta modulaci, a to až 24 bitů. Nevýhodou je použití pouze pro nižší frekvence, typicky do desítek kHz. [23]

3.7.3 Výběr A/D převodníku

A/D převodník bude sloužit k převodu stejnosměrných napětí, které se budou vzhledem k vzorkovacím frekvencím převodníku měnit jen velice pomalu. Není zde tedy praktický limit pro vzorkovací kmitočet.

Důležitým parametrem je počet kvantizačních hladin převodníku. Z výpočtů (10), (11) a (16) je patrné, že minimální délka výstupního bitového slova A/D převodníku je 9 bitů. Pro ještě lepší měření by bylo vhodné použít převodník s délkou bitového slova 12 či více bitů. Potřeba jsou minimálně 3 vstupní kanály.

Požadavkem na A/D převodník je také komunikace přes sběrnici I²C nebo SPI, aby mohl být připojen k čipu i.Mx8DV.

Výše popsané požadavky splňuje například A/D převodník AD7991 od firmy Analog Devices. Jedná se o 12 bitový převodník, se čtyřmi vstupními kanály. Napájecí napětí 2,7 až 5,5 V. Převodník je vyráběn v pouzdře SOT-23.

4 KOMUNIKACE PO SÍTI

Platforma i.Mx8DV obsahuje dva ethernetové porty a neobsahuje žádné bezdrátové rozhraní. Proto je v této práci řešena pouze komunikace pomocí ethernetu.

4.1 Ethernet

Síť typu Ethernet využívá tzv. sběrnicovou topologii. Jedná se o topologii, kde je dané přenosové médium sdíleno společně všemi zařízeními připojenými k tomuto médium. Všechna zařízení před vysláním monitorují provoz v síti. K vyslání své vlastní zprávy dojde až v případě, kdy je médium (například síťový kabel) prázdné. Dojde-li k vyslání dvou stanic ve stejném čase, jejich pakety se střetnou (dojde ke kolizi) a obě stanice vyslání přerušují. Každá stanice počká náhodně dlouhý interval po kterém dojde k opětovnému vyslání. Pro přístup ke společnému přenosovému médium je využita metoda CSMA/CD (Carrier Sence with Multiple Access and Collision Detection). Tato metoda řídí naslouchání síťového provozu, detekci kolizí, přerušení, atp. Po úspěšném odeslání zprávy uzlem do sítě obdrží vysílanou zprávu všechny stanice. Každá stanice vyhodnotí, je-li cílová adresa stejná jako její vlastní. Pokud se shodují, zařízení zprávu přijme a zpracuje. Pokud se neshodují, zařízení zprávu zahodí. Každé zařízení je v síti identifikováno svou vlastní hardwarovou adresou (adresa MAC).[35]

O standardizaci Ethernetu pro veřejné použití se staral IEEE. Standardy IEEE, zabývající se Ethernetem dělíme do tří skupin následovně. [34]

4.1.1 Klasický Ethernet

V dnešní době se již téměř nepoužívá. Maximální přenosová rychlost je 10 Mb/s. Mezi klasický Ethernet se řadí například typy 10Base2, 10Base5, 10BaseT nebo 10BaseFL, u kterého je poprvé použit optický kabel. Typická je sběrnicová topologie sítě. [34]

4.1.2 Fast Ethernet

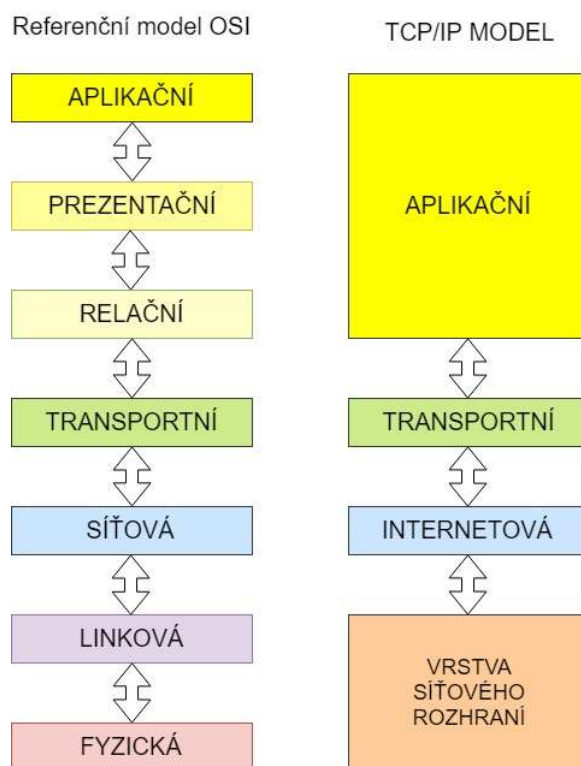
Fast Ethernet je dnes asi nejpoužívanějším typem Ethernetu, zejména v domácích sítích. Maximální přenosová rychlost 100 Mb/s. Topologie sítě bývá nejčastěji hvězdicová a využívá kabely UTP (Unshielded Twisted Pair) nebo STP (Shielded Twisted Pair), případně optické kabely. [34]

4.1.3 Gigabit Ethernet

Maximální přenosová rychlost větší nebo rovna 1Gb/s. Pro kratší vzdálenosti lze použít i kabely UTP, většinou se však používají optické a STP kabely.

4.2 Síťový model OSI

Referenční model OSI je standard vyvinutý v roce 1984 společností ISO za účelem sjednocení způsobu spolupráce komponent síťových protokolů. Tento model rozděluje síťové funkce do sedmi vrstev (viz Obrázek 20: Porovnání modelů OSI a TCP/IP) a definuje, jak se data vytvořené v aplikaci počítače A převedou pomocí síťového média do aplikace počítače B. Jednotlivé vrstvy modelu OSI představují logické seskupení postupných úkonů, které je potřeba provést při komunikaci. [34]



Obrázek 20: Porovnání modelů OSI a TCP/IP

4.2.1 Vrstvy OSI

Fyzická vrstva - data mají na této vrstvě podobu signálu (ať už elektrického nebo optického) a jsou přenášena v binární podobě. Fyzická vrstva definuje fyzikální vlastnosti sítě, například napěťové úrovně pro jednotlivé bity, maximální délku přenosového média, použitou frekvenci apod.

Linková vrstva - někdy také označována jako datová nebo spojová. Linková vrstva upravuje data, které do ní vstupují z vrstev nad ní, pro cestu přenosovým kanálem. Hovoříme zde o fyzickém adresování, detekci chyb, řízení přístupu k fyzickému přenosovému médium. Řešíme zde vlastní topologii sítě. [34]

Síťová vrstva - má za úkol řadit pakety určené k odeslání. Tato vrstva je důležitá z hlediska výběru optimální trasy paketů v síti. Z toho důvodu musí zjišťovat fyzické adresy příjemců i dalších zařízení, přes které pakety směřují. V síťové vrstvě rozpoznáváme protokol IP. [34]

Transportní vrstva - v transportní vrstvě se logicky řeší transport, resp. přenos, dat po

jednotlivých uzlech. Jedním z protokolů pracujících v transportní vrstvě je protokol TCP (viz obrázek 20), pomocí něhož řídí tok dat od jednoho uzlu k jinému, přičemž provádí kontrolu jestli byl přenos kompletní. Pokud tomu tak není, je na této vrstvě možné přenos opakovat díky preventivnímu ukládání odesílaných dat. Mezi další protokoly na transportní vrstvě patří například UDP nebo DCCP.

Další důležitou funkcí je také logická adresace portů. Což jsou čísla aplikací v přijímajícím zařízení, kterým je sada paketů určena. Bez adresace portů by příjemce jen složitě rozpoznával, která aplikace ten který paket vyžaduje. [34]

Relační vrstva je odpovědná za vytvoření, sledování a následné ukončení komunikační relace mezi dvěma komunikujícími stanicemi. Poté, co je vytvořena sada paketů k odeslání, naváže spojení odesílatele s příjemcem. Nejdůležitější akcí relační vrstvy je vytváření hlaviček paketů, které obsahují informace o začátku a konci zprávy.

Hlavička také obsahuje informace o tom, jakým způsobem bude komunikace probíhat. Na výběr jsou dva způsoby, Full-Duplex a Half-Duplex, způsob je vybrán také na relační vrstvě.[34]

Full-Duplex (plně duplexní) znamená, že komunikace mezi odesílatelem a příjemcem může probíhat současně oběma směry. Při použití Half-Duplex (poloviční duplex) je komunikace možná vždy jen jedním směrem. Druhá stanice čeká až vysílání skončí a pak se směr vysílání otočí.

Prezenční vrstva - tato vrstva zajišťuje překlad dat do formy, která je srozumitelná pro určenou aplikaci. Jelikož se data při odesílání kódují, šifrují nebo komprimují, je do hlavičky paketu připsána informace o použité metodě komprese, šifrování nebo kódování. Podle toho je prezentační vrstva schopna přetransformovat zprávu do původní formy. Procesy probíhající na této vrstvě jsou samozřejmě obousměrné. A to podle toho, jestli zprávu vysílá nebo přijímá. [34]

Aplikační vrstva - účelem aplikační vrstvy je poskytnout aplikacím přístup ke komunikačním prostředkům. Vytvoří-li aplikace data, která by chtěla poslat, pak tyto data předá aplikační vrstvě.

4.3 Síťový model TCP/IP

Koncepce modelu ISO OSI není jedinou koncepcí, která umožňuje síťovou komunikaci. V současné době je nejvíce používaný model TCP/IP. Model TCP/IP není složen jen z protokolu TCP a IP, ale jedná se o celou sadu protokolů. Koncepce TCP/IP je rozdělena jen do čtyř vrstev [34] (viz Obrázek 20: Porovnání modelů OSI a TCP/IP)

Vrstva síťového rozhraní - tato vrstva má v modelu TCP/IP na starosti souhrn veškerých činností souvisejících s ovládáním dané přenosové cesty, odesílání a přijímání paketů. Často se také můžeme setkat s označením Ethernetová vrstva. [34]

Internetová vrstva - Internetová, popřípadě síťová vrstva je realizována protokolem IP a její funkce je stejná jako funkce síťové vrstvy v modelu OSI.

Transportní vrstva - pracuje s protokoly TCP a UDP podle toho, jedná-li se o zabezpečený nebo nezabezpečený přenos dat. Vrstva rozděluje data na segmenty a případně kontroluje jejich správnost doručení.

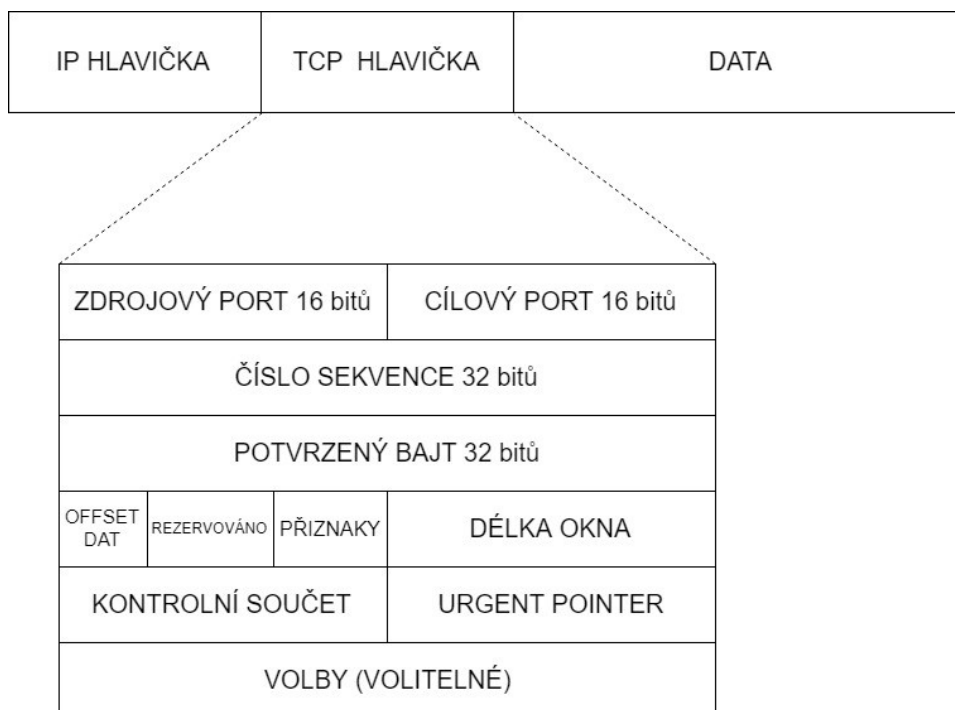
Aplikační vrstva - aplikační vrstvu tvoří hned několik protokolů, které spolupracují s jednotlivými aplikacemi. Například protokol HTTP sloužící mimo jiné pro komunikaci internetových prohlížečů s webovými servery.

4.4 TCP

Protokol TCP (transportní vrstva) je protokol vyšší vrstvy než protokol IP (síťová vrstva). TCP dopravuje data mezi dvěma konkrétními aplikacemi, zatímco IP mezi libovolnými počítači v internetu. Protokol TCP využívá k transportu dat internetem protokol IP, avšak nad tímto protokolem zřizuje spojovanou službu. Tj. že mezi dvěma aplikacemi naváže spojení. Vytvoří na dobu spojení virtuální okruh, který je plně duplexní - data se přenášejí současně na sobě nezávisle oběma směry. Musí řešit problémy navázání a ukončení spojení, potvrzování přijatých dat (přenášené bajty jsou číslovány, integrita přenášených dat je zabezpečena kontrolním součtem), vyžádání ztracených dat (ztracená nebo poškozená data jsou znovu vyžádána), ale také problémy průchodnosti přenosové cesty. Aplikace používající protokol TCP si nemusí dělat starosti s tím, zdali náhodou nebyla nějaká data během přenosu ztracena nebo díky chybě přenosu pozměněna. [35]

4.4.1 TCP Hlavička

Cílová aplikace je v internetu jednoznačně určena IP adresou, číslem portu a použitým protokolem (TCP nebo UDP). Protokol IP dopraví IP datagram na konkrétní počítač, kde běží jednotlivé aplikace a podle čísla cílového portu operační systém pozná, které aplikaci má TCP segment doručit. Základní přenosovou jednotkou protokolu TCP je TCP segment. TCP segment se skládá z TCP hlavičky a přenášených dat. Vkládá se do IP datagramu. IP datagram zase do linkového rámce. [35]



Obrázek 21: TCP hlavička

4.4.2 Komunikace protokolem TCP

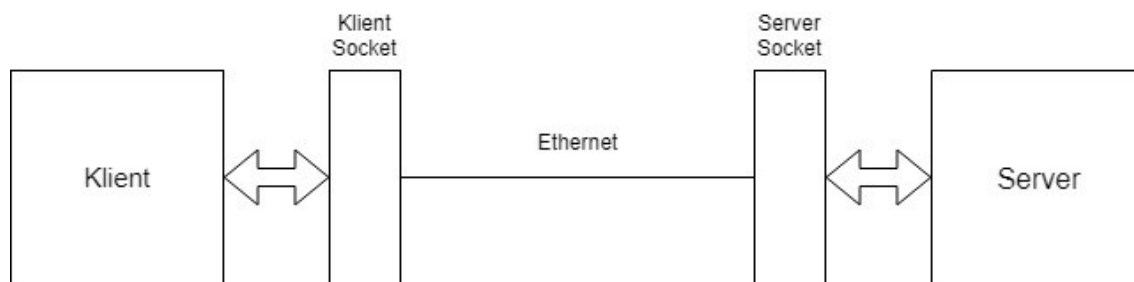
Komunikace popisuje výměnu segmentu TCP mezi oběma konci TCP spojení.

Navázání spojení začíná klient odesláním prvního TCP segmentu na port serveru. Klient vygeneruje náhodné číslo v rozsahu od 0 do $2^{32} - 1$, které použije jako startovací číslo odesílaného bajtu (tzv. ISN). Vytvořením startovacího čísla bajtu se nastaví v segmentu příznaků SYN. Takovýto segment je prvním segmentem v TCP komunikaci, proto nemůže potvrzovat žádná přijatá data. Nemůže být tedy nastaven příznak ACK. Fakt, že segment nese aplikační data je vyjádřen nastavením příznaku PSH.[35]

Ukončení spojení může generovat kterákoli strana odesláním TCP segmentu s příznakem FIN, tzv. aktivním ukončením spojení. Druhá strana reaguje tzv. pasivním ukončením spojení. Pro patřičné ukončení jsou nutné 4 TCP segmenty. [35]

4.5 Socket

Na socket můžeme nahlížet jako na abstraktní meziprocesní komunikační prostředek, který je definovaný jako koncový kanál. Socket tedy obecně definuje kombinaci IP adresy (identifikace počítače) a portu (identifikace procesu). Socket si lze také představit i jako určitou datovou strukturu nebo objekt, který nese informace o stavu síťové komunikace s druhou stranou. Socketový mechanismus se nejvíce používá pro typ komunikace pracující na bázi klient-server. [39] Tento mechanismus je zobrazen na obrázku 22.



Obrázek 22: Zobrazení principu socketů na bázi klient-server

4.6 HTTP protokol

Hypertext Transfer Protocol (HTTP) je aplikační komunikační protokol pro přenos informací na Internetu. Používá se k přístupu na WWW stránky a související data. Je to tedy zejména protokol pro komunikaci WWW prohlížeče, který stránky stahuje a zobrazuje uživateli, s WWW serverem, který stránky poskytuje. [36]

4.6.1 Požadavek klienta

Obvyklé použití protokolu HTTP spočívá v tom, že uživatel zadá ve svém prohlížeči do adresního řádku URL adresu stránky, která ho zajímá. WWW prohlížeč si z této URL adresy vezme název serveru, který přeloží na IP adresu. Prohlížeč pak otevře spojení s daným serverem, který na svém vstupu očekává požadavek. Klient zašle serveru svůj požadavek a server na něj odpoví. Komunikace probíhá nad protokolem TCP většinou na standardním portu 80. [36]

Na prvním řádku požadavku se uvádí metoda (požadovaná operace), cesta (URL adresa) a označení použitého protokolu. Metoda říká, jaká operace se má na serveru provést. Metod, které definuje HTTP/1.1 je mnoho, ale nejvíce se používanými metodami jsou [36]:

- GET – získání objektu (souboru) s danou URL
- POST – odeslání dat z formuláře a získání objektu s danou URL
- HEAD – získání objektu s danou URL bez těla (obdrží pouze hlavičku odpovědi)

Následuje cesta, resp. Část URL adresy, udávající cestu k objektu na WWW serveru. Na dalších řádcích, kterých může být libovolné množství, se uvádějí další hlavičky, tedy dodatečné informace, které ovlivňují zpracování požadavku nebo formát odpovědi. [36]

4.6.2 Odpověď serveru

Odpověď serveru má velmi podobný formát. Na prvním řádku je nejprve uvedena verze http protokolu a uveden trojčíselný kód, označující výsledek operace, popř. číslo chyby, a dále textový popis významu kódu. Obvyklé kódy jsou:

- 1xx – informační hlášky – téměř nepoužívané
- 2xx – úspěch
 - 200 OK – požadavek proběhl v pořádku
- 3xx – přesměrování (liší se důvody)
- 4xx – chyba na straně klienta – v požadavku je chyba nebo nemůže být vyřízen
- 5xx – chyba na straně serveru

Na dalších řádcích se mohou vyskytovat hlavičky, které doplňují informace k odpovědi. [36]

4.7 HTML

HyperText Markup Language (HTML) je značkovací jazyk vytvořený pro hypertext. Je jedním z rodiny jazyků pro vytváření stránek v systému WWW, který umožňuje publikaci stránek na Internetu. Je charakterizován množinou tagů (značek) a jejich atributů pro danou verzi definovaných. Mezi značky se uzavírají části textu dokumentu a tím se určuje význam obsaženého textu. [37]

5 SOFTWAREVÉ ŘEŠENÍ

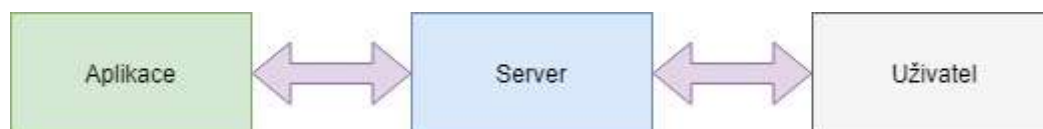
Cílem této kapitoly je přiblížit možné a vytvořené softwarové řešení inteligentního systému pro správu akvária běžící na systému QNX Neutrino a vývojové desce i.Mx8DV.

5.1 Možná řešení

Možností, jak vytvořit systém splňující všechny požadavky, je několik. Ze zadání této práce je patrné, že systém musí poskytovat grafické uživatelské rozhraní a blíže neurčeným způsobem musí umožňovat vzdálenou správu pomocí webového rozhraní. A právě propojení těchto dvou požadavků lze provést několika způsoby.

5.1.1 Vzdálený server

Jednou z možností jak zajistit vzdálenou správu je použití externího serveru. Aplikace odesílá data v pravidelných intervalech přes internet na vzdálený server. Ten data zpracuje a je připraven poskytnout je uživateli až o ně požádá. V případě, že chce uživatel změnit nastavení systému zašle nové nastavení na server a ten požadavek předá aplikaci. Tento princip je zobrazen na obrázku 23. Fialové šipky znázorňují komunikaci pomocí protokolu HTTP.



Obrázek 23: Princip komunikace přes vzdálený server

Výhodou tohoto řešení je možnost přistupovat k datům odkudkoliv. Nevýhodou může být nutnost připojit aplikaci do sítě internet a případné poplatky spojené s provozem serveru.

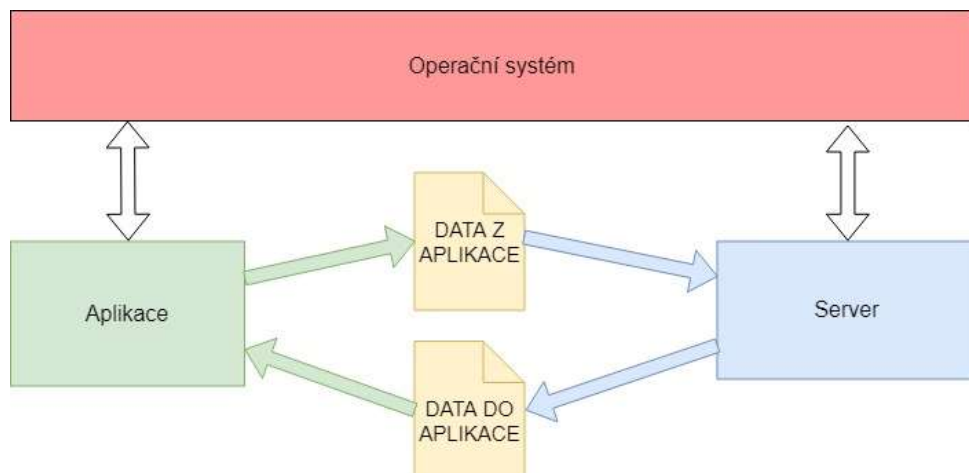
5.1.2 Server mimo aplikaci

Další možností je spustit server přímo na zařízení i.Mx

8 pod operačním systémem QNX. Tím odpadá potřeba používat vzdálený server a data jsou přístupná i po lokální síti. Vhodný pro účely webového serveru je například modul CGIHTTPServer jazyka Python. Tento modul spustí při příchodu požadavku skript, který vygeneruje webovou stránku a ta je poté serverem odeslána. Nevýhodou tohoto řešení je nutnost implementace vzájemného rozhraní mezi aplikací a serverem.

Nejjednodušší je jako vzájemné rozhraní použití souboru. V tom případě aplikace zapisuje data do souboru. Při příchodu požadavku pro čtení na server je spuštěn skript, který soubor otevře a na základě dat ze souboru vygeneruje HTML stránku. Ta je pak odeslána uživateli. Při požadavku na zápis je spuštěn jiný skript, který přijatá data zapíše

do jiného souboru, který aplikace v pravidelném intervalu čte a zjišťuje jestli je k dispozici nové nastavení. Dva soubory jsou nutné pro zabránění kolize v současném zapisování do jednoho souboru. Tento princip je zobrazen na obrázku 24. Zelené šipky představují pravidelný zápis a čtení prováděné aplikací. Modré šipky přenos dat jen při příchodu požadavku na server.



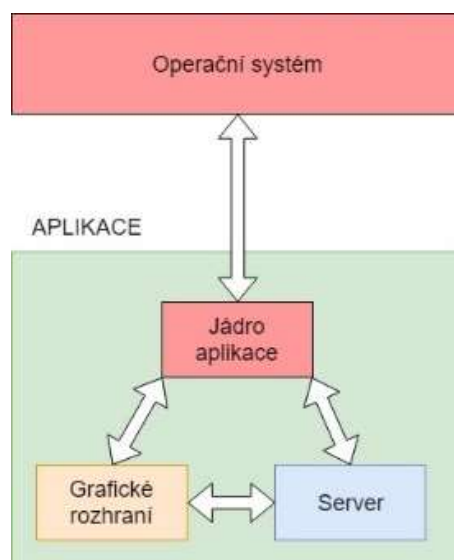
Obrázek 24: Princip výměny informací pomocí souborů

Nevýhodou tohoto řešení je neustálé zapisování do souboru a prodleva při zápisu dat směrem od uživatele k aplikaci. Tato prodleva je dána frekvencí čtení souboru. Délka prodlevy v řádu jednotek sekund by však neměla zásadní vliv na funkčnost a uživatelskou přívětivost systému správy akvária.

5.1.3 Server v aplikaci

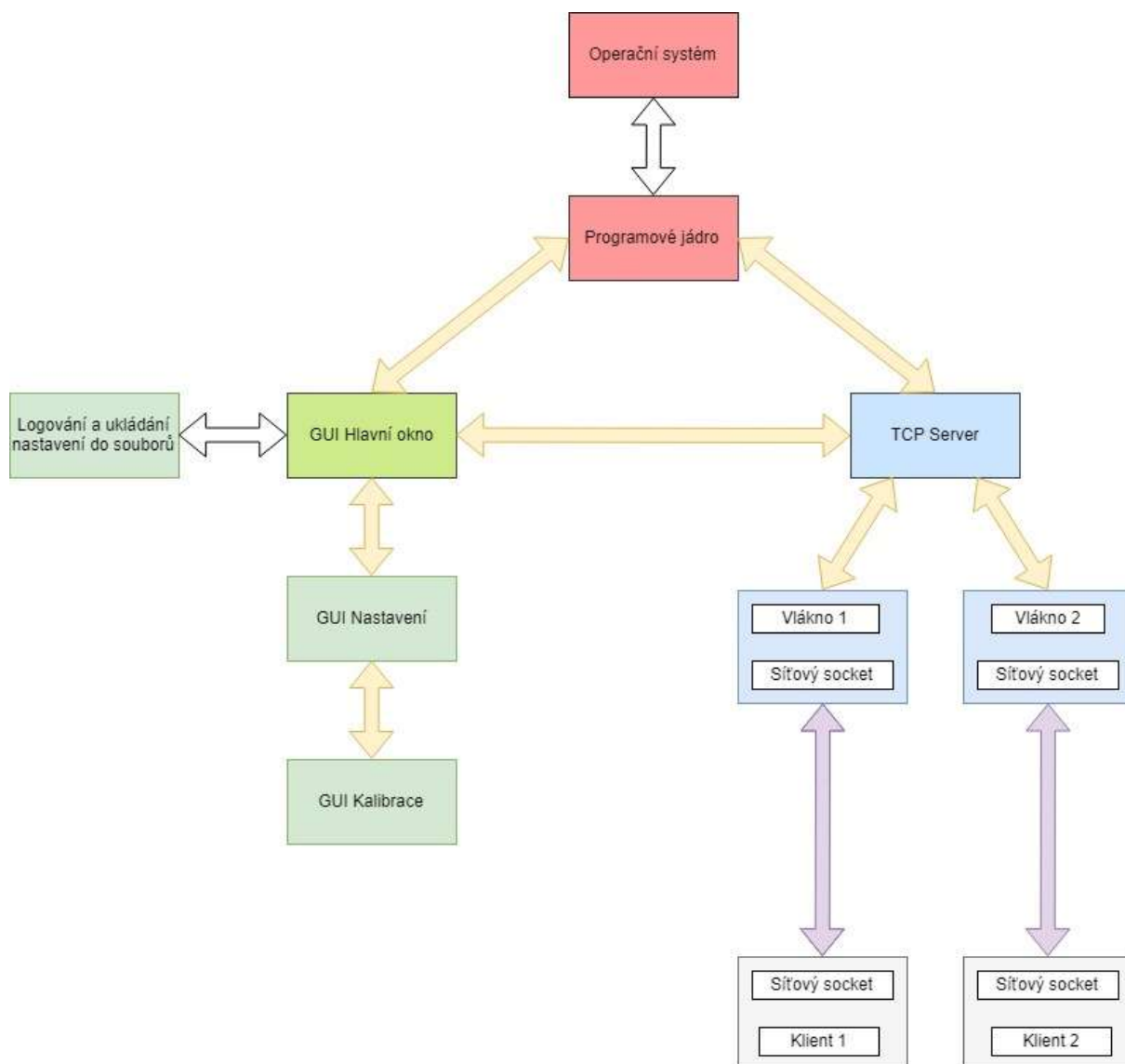
Poslední zde uváděná možnost je integrace webového serveru přímo do aplikace. Do projektu s aplikací je potřeba přidat Qt modul *network*, který poskytuje třídy potřebné pro síťové programování. Zejména třídu *QTcpSocket* a *QTcpServer*. S touto metodou odpadá potřeba zajišťovat komunikaci mezi serverem a aplikací jako takovou. Řeší se jen komunikace uvnitř aplikace mezi objektem serveru s ostatními objekty uvnitř aplikace, což lze snadno implementovat pomocí signálů a slotů. Tento princip je zobrazen na obrázku 25.

Tento způsob připojení serveru k aplikaci byl na základě jeho předností oproti ostatním způsobům použit při tvorbě aplikace pro řízení systému správy akvária.



Obrázek 25: Princip serveru v aplikaci

5.2 Architektura vytvořeného systému



Obrázek 26: Zobrazení architektury aplikace

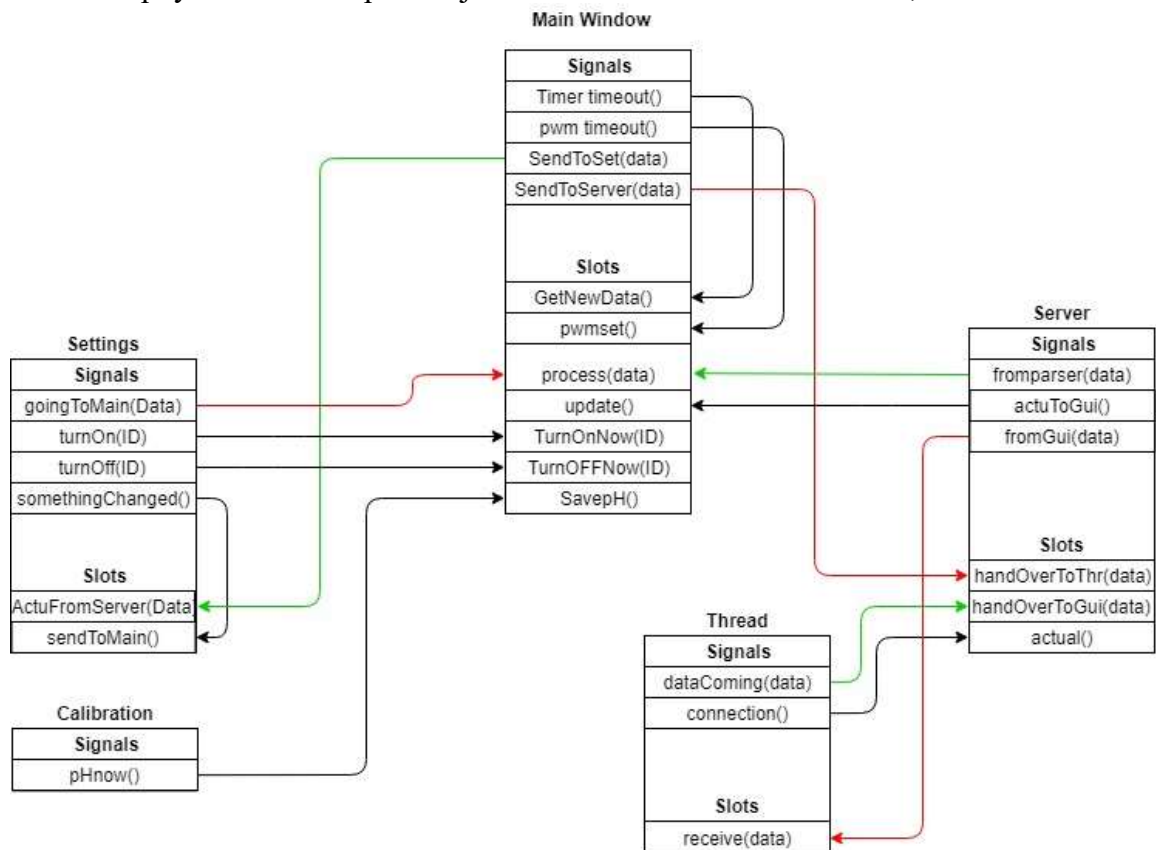
Na obrázku 26 je zobrazena architektura vytvořené aplikace pro správu akvária. Programové jádro při startu vytváří objekty *Hlavní grafické okno* (viz 5.4.1) a *Server TCP* (viz 4). Uživatel může v hlavním grafickém okně otevřít další okno *Nastavení* (viz 5.4.2) a z něj pak vyvolat okno *Kalibrace* (viz 5.4.3). Žluté šipky představují komunikaci pomocí signálů a slotů uvnitř aplikace. Fialové šipky znázorňují komunikaci pomocí protokolu HTTP nad protokolem TCP. Klient 1 a klient 2 představují webové prohlížeče, které jsou připojeny k aplikaci za účelem vzdálené správy systému. Hlavní grafické okno zajišťuje logování informací o systému při běhu aplikace. Stará se také o ukládání a nahrávání nastavení při ukončení a zapnutí aplikace.

5.3 Realizace

V této podkapitole je popsána komunikace pomocí signálů a slotů mezi jednotlivými objekty v aplikaci a použité algoritmy.

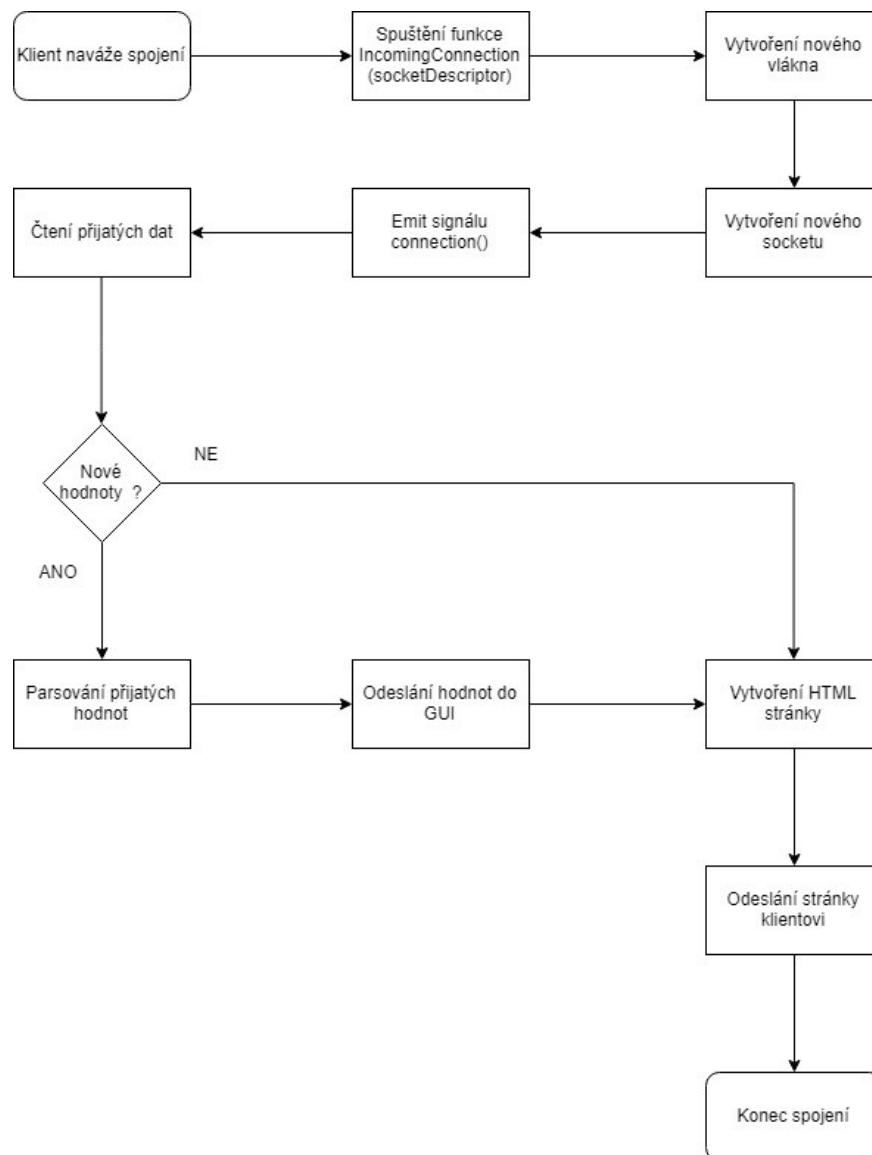
5.3.1 Přehled signálů a slotů

Na obrázku 27 jsou zobrazeny z důvodu větší přehlednosti jen nejdůležitější signály a sloty použité v aplikaci. Červené šipky znázorňují předávání dat směrem z grafického rozhraní na server. Zelené šipky směr opačný, tedy ze serveru do grafického rozhraní. Černé šipky většinou přenášejí informaci o určité změně, která nastala.



Obrázek 27: Zobrazení signálů a slotů v aplikaci

5.3.2 Algoritmus komunikace klienta se serverem



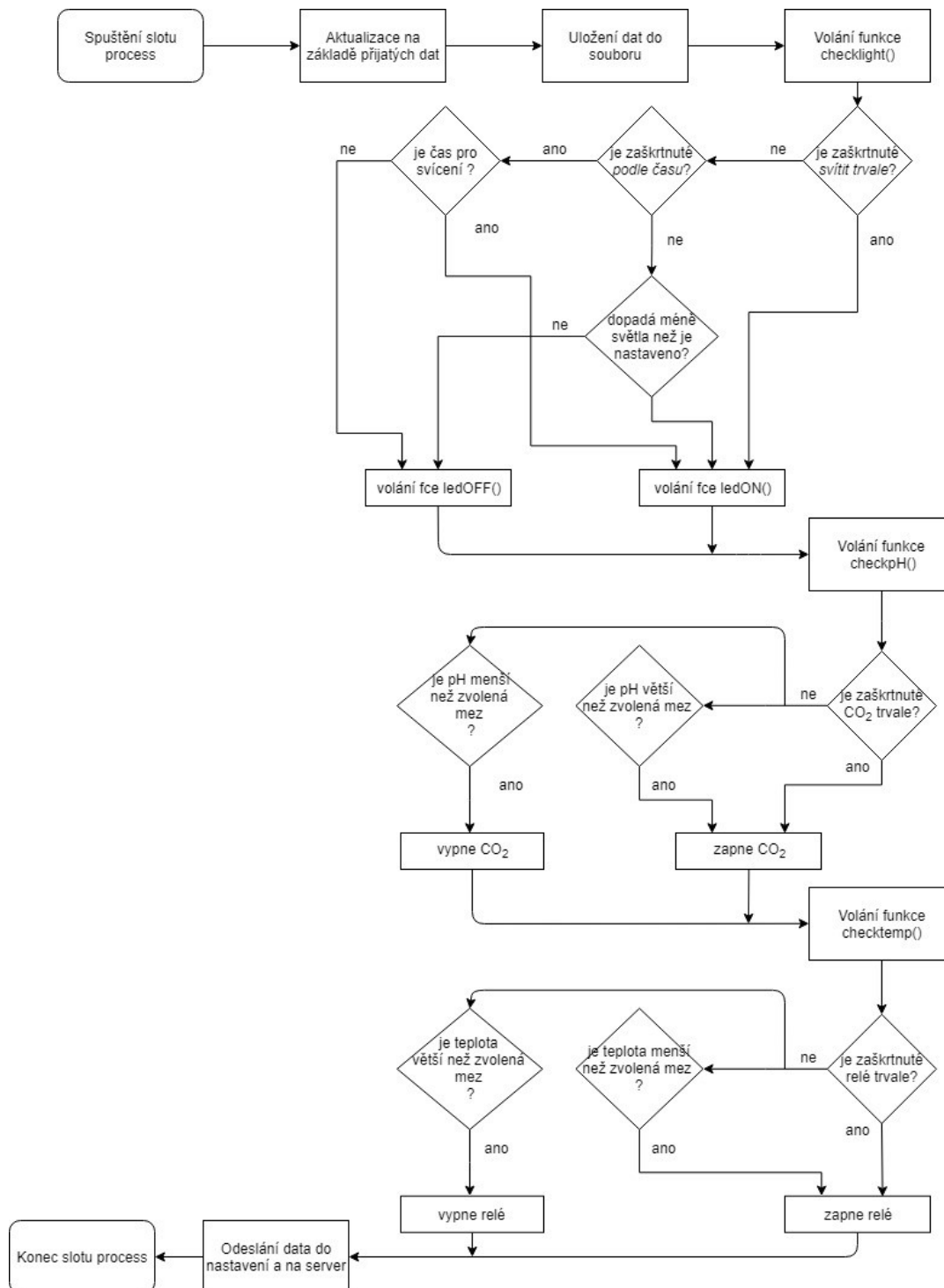
Obrázek 28: Algoritmus obsluhy požadavku klienta

Při navázání komunikace se spustí funkce *IncomingConnection*, která pro každé spojení vytvoří nové vlákno. Díky tomu může být v jeden okamžik k serveru připojeno několik klientů. V praxi by při nasazení tohoto systému mohl být použit i jednovláknový server, protože je velice nepravděpodobné, že by ve firmě s několika málo desítkami zaměstnanců došlo ke dvěma a více současným připojením.

Dále je emitován signál *connection*. Jeho vyvolání zajistí aktualizaci dat z GUI na server. Následuje načtení dat, které klient poslal. Pokud byla přijata nová nastavení, dojde k jejich parsování dat a odeslání hodnot do GUI. Následuje vytvoření HTML stránky a odeslání stránky klientovi. Spojení je ukončeno.

5.3.3 Algoritmus změny nastavení

Na obrázku 29 je zobrazen zjednodušený algoritmus zpracování dat při jejich změně. Nezáleží jestli změnu vyvolat uživatel přes GUI nebo webové rozhraní.



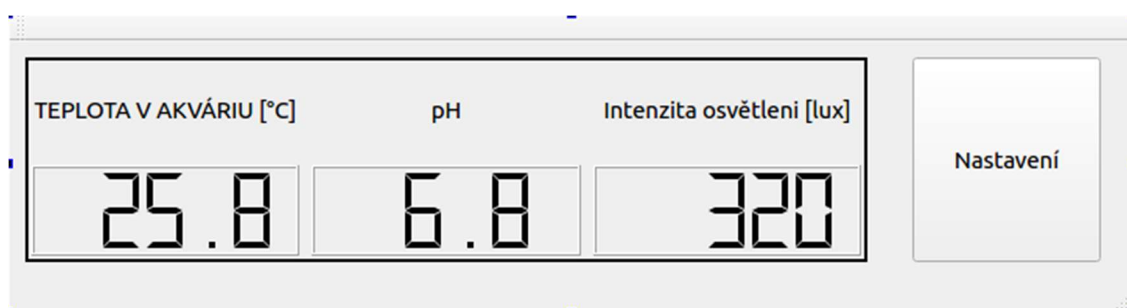
Obrázek 29: Algoritmus pro zpracování dat při jejich změně

5.4 Popis vytvořeného grafického uživatelského rozhraní

Grafické uživatelské rozhraní (GUI) slouží k interakci uživatele s aplikací. Na pozadí tohoto grafického rozhraní je aplikace, která se stará o chod celého systému správy akvária. Všechna nastavení, které uživatel provede, jsou uložena a při dalším spuštění je aplikace uvedena do stejného stavu jako před vypnutím. Uživatel tak při každém startu systému nemusí zadávat parametry znovu.

5.4.1 Hlavní grafické okno

Na obrázku 30 je zobrazena úvodní obrazovka GUI. Zobrazuje nejdůležitější měřené parametry. Teplotu vody v akváriu ve °C, pH vody a intenzitu osvětlení. V pravé části je umístěno tlačítko *Nastavení*, které otevírá další obrazovku na které lze nastavit vše potřebné.



Obrázek 30: Úvodní obrazovka GUI

5.4.2 Obrazovka nastavení

V horní části obrazovky se nachází pět číselných přírůstkových polí (spin box). Pomocí těchto polí lze nastavit požadovanou hodnotu teploty a maximální přípustnou odchylku. Stejně tak lze nastavit chtěnou hodnotu pH a maximální odchylku. Páté přírůstkové pole slouží k zadání hodnoty osvětlení při které se sepne osvětlení.

Dále se na obrazovce nachází tři zaškrťovací pole. Při zaškrtnutí pole *sepnout trvale relé* dojde k trvalému sepnutí relé. To dává uživateli možnost do takto spínané zásuvky připojit například motorek na vzduchování. Protože primárně je relé určené ke spínání topného tělesa dojde při zaškrtnutí pole *sepnout trvale relé* k přerušení funkce regulace teploty. Teplota je nadále měřena a zobrazována, ale je znemožněno zadávat hodnoty pro nastavení teploty. Této funkce může uživatel využít například pokud má v akváriu svůj vlastní systém ohřevu vody.

Zaškrťovací pole *sepnout trvale CO2* má obdobnou funkci jako *sepnout trvale relé*. Při zaškrtnutí tohoto pole dojde k trvalému sepnutí ventilu na CO₂. Při neustálém dodávání CO₂ do akvária systém nemůže regulovat pH vody. Proto znemožní uživateli zadávat požadovanou hodnotu a hysterezi pH. I nadále je pH měřeno a zobrazováno. Tuto funkci může uživatel využít v případě akvárií s velkým výskytem rostlin, kde je důležitější dostat do akvária dostatečné množství CO₂ než přesně regulovat hodnoty pH vody. Třetí zaškrťovací pole *sepnout trvale osvětlení* při zaškrtnutí trvale zapne osvětlení.

a nehledí přitom na prahovou hodnotu osvětlení ani na časové rozmezí ve kterém má být osvětlení zapnuto.

	požadovaná hodnota	hystereze	od	do
teplota [°C]:	25.60	1.80	23.8	27.4
pH [-]:	7.00	0.50	6.5	7.5
osvětlení [lux]:	0			

sepnout trvale: ☐ relé ☐ CO2 ☐ osvětlení

OSVĚTLENÍ

řídít podle:

☐ prahové hodnoty ☒ času

rozsvítit v: 7:00 zhasnout v: 19:00

Kalibrovat pH sondu Zpět

Obrázek 31: Obrazovka s nastavením

Ve spodní polovině obrazovky se nachází část pro nastavení spínání osvětlení. V levé části je umístěn prvek umožňující uživateli vybrat si podle čeho se má spínání osvětlení řídit. Na výběr jsou dvě možnosti. Při zvolení první z nich, tedy *prahové hodnoty* se osvětlení sepne vždy, když osvětlení na fotorezistoru klesne pod nastavenou hodnotu. Při zvolení druhé možnosti tedy *času* je osvětlení spínané podle nastaveného času. Aplikace pro tyto účely využívá systémový čas operačního systému.

5.4.3 Kalibrace sondy

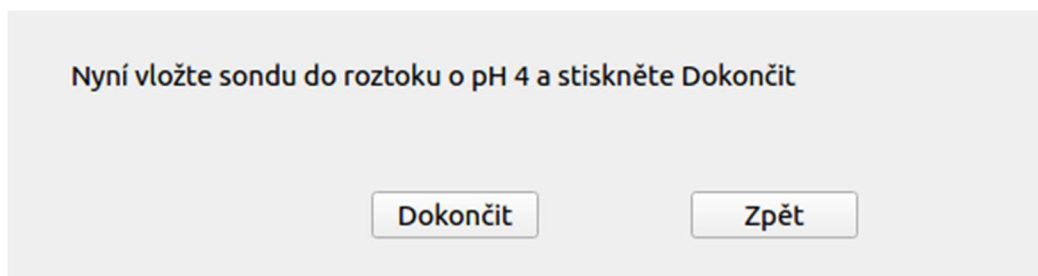
Sonda pro měření pH vody by měla být přibližně jednou za měsíc kalibrována. Pro kalibraci se používají pufry. Jedná se o roztoky, které mají stabilní, přesně definovanou hodnotu pH. Pro kalibraci sondy jsem zvolil běžně nejdostupnější pufry, tedy hodnoty pH 7 a 4.

Vložte sondu do roztoku o pH 7 a stiskněte tlačítko Další

Další Zpět

Obrázek 32: Kalibrace sondy krok 1

Při kliknutí na tlačítko *Kalibrovat pH sondu* se otevře okno, které vyzve uživatele k vložení sondy do roztoku o pH 7. Po stisknutí tlačítka *Další* je uživatel vyzván, aby nyní vložil sondu do roztoku o pH 4.



Obrázek 33: Kalibrace sondy krok 2

Po stisknutí tlačítka *Dokončit* je kalibrace hotova, okno se zavře a uživatel je nyní zpět v okně s nastavením.

Aplikace v intervalu jedné sekundy průměruje hodnoty získané z AD převodníku (teplota, pH vody a intenzita osvětlení) a na jejich základě se rozhoduje jestli je potřeba sepnout či rozepnout dané výstupní obvody, případně nedělat nic.

5.5 Webové rozhraní

Webové rozhraní slouží k interakci uživatele s aplikací pomocí vzdáleného přístupu. Při zadání IP adresy a příslušného portu zařízení na kterém běží systém správy akvária do webového prohlížeče se uživateli zobrazí stránka html s aktuálním stavem akvária a s možností nastavit požadované parametry.

Na začátku webové stránky jsou zobrazeny aktuální informace o stavu vody v akváriu.

Následuje sekce *Nastavení*. Stejně jako v grafickém uživatelském rozhraní začíná tato sekce možností nastavit požadovanou teplotu, pH a osvit. Následují tři zaškrťovací pole, kde uživatel může zvolit má-li být zvolený výstupní obvod sepnutý trvale. Dále je potřeba zvolit podle čeho se má řídit osvětlení. Uživatel si stejně jako na dotykové obrazovce může vybrat jestli má být osvětlení řízeno podle prahové hodnoty nebo podle času. Sekci *Nastavení* uzavírá možnost nastavení času v kterém má být osvětlení akvária zapnuto a vypnuto. Jako úplně poslední prvek je tlačítko *Nastavit*, pomocí kterého se provedené nastavení odešle do zařízení a příslušné parametry se přenastaví. Stránka se aktualizuje automaticky po 10 sekundách a při každém odeslání nového nastavení.

akvarium x +

← → ↻ 🏠

🔍 192.168.0.2:8000

Vzdálený přístup pro správu akvária

Aktualní hodnoty:
teplota: 25.2
pH: 7
světlo: 200

Nastavení:

teplota:	<input type="text" value="26"/>	hystereze	<input type="text" value="0,5"/>
pH:	<input type="text" value="7,2"/>	hystereze:	<input type="text" value="0,4"/>
osvětlení:	<input type="text" value="300"/>		

sepnout trvale:

relé ☐

CO₂ ☒

světlo ☐

osvětlení řídit podle:

☒ času

☐ prahové hodnoty

zapnout světlo v:

vypnout světlo v:

Obrázek 34: Zobrazení webového rozhraní

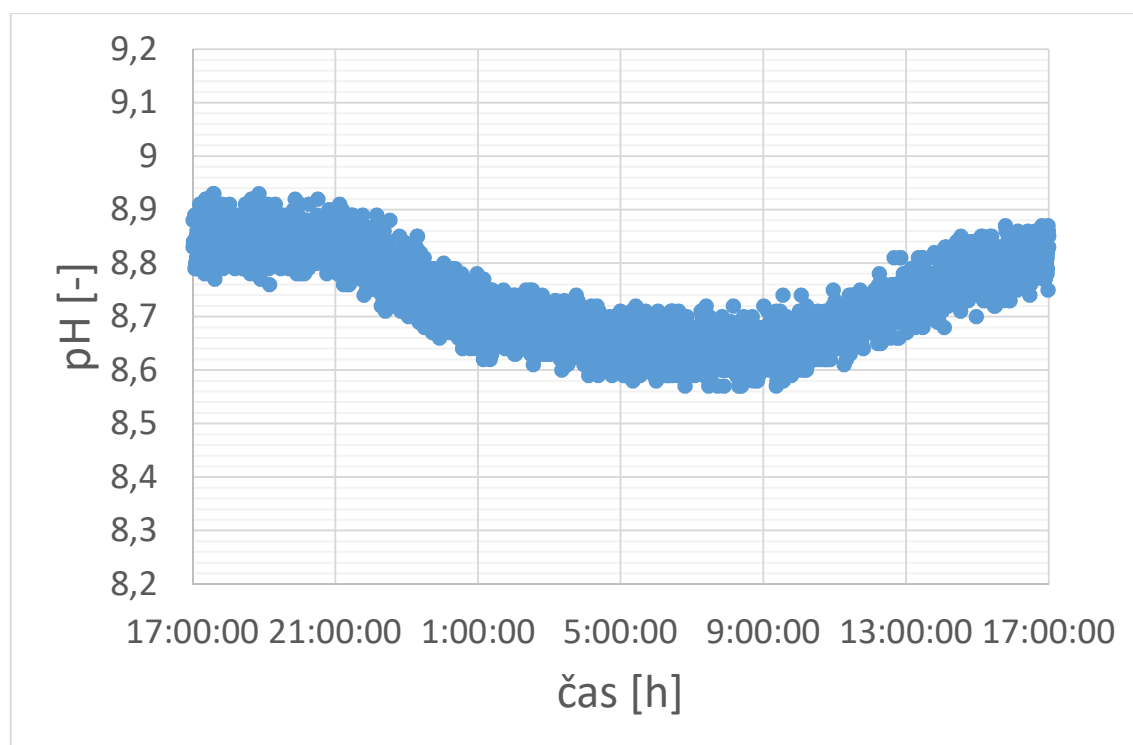
6 TESTOVÁNÍ ZAŘÍZENÍ

V této kapitole jsou shrnuty poznatky a data získané při testování zařízení.

6.1 Naměřená data

6.1.1 Bez dodávání CO₂

Na obrázku 35 je zobrazen průběh hodnot pH vody v akváriu během dne pokud do akvária není dodáváno CO₂. Testování probíhalo v akváriu s rostlinami. Objem akvária je 100 l. V akváriu bylo od 21 do 9 hodin vypnuto osvětlení. Před a po tomto čase bylo akvárium osvětleno dvěma led trubicemi, každá o výkonu 16 W. V akváriu byl umístěn akvarijní filtr, který byl během měření spuštěný nepřetržitě. Deklarovaným průtok filtru je 500 l/hod. Měření probíhalo 24 hodin. Teplota vody se pohybovala od 25 do 27 °C.

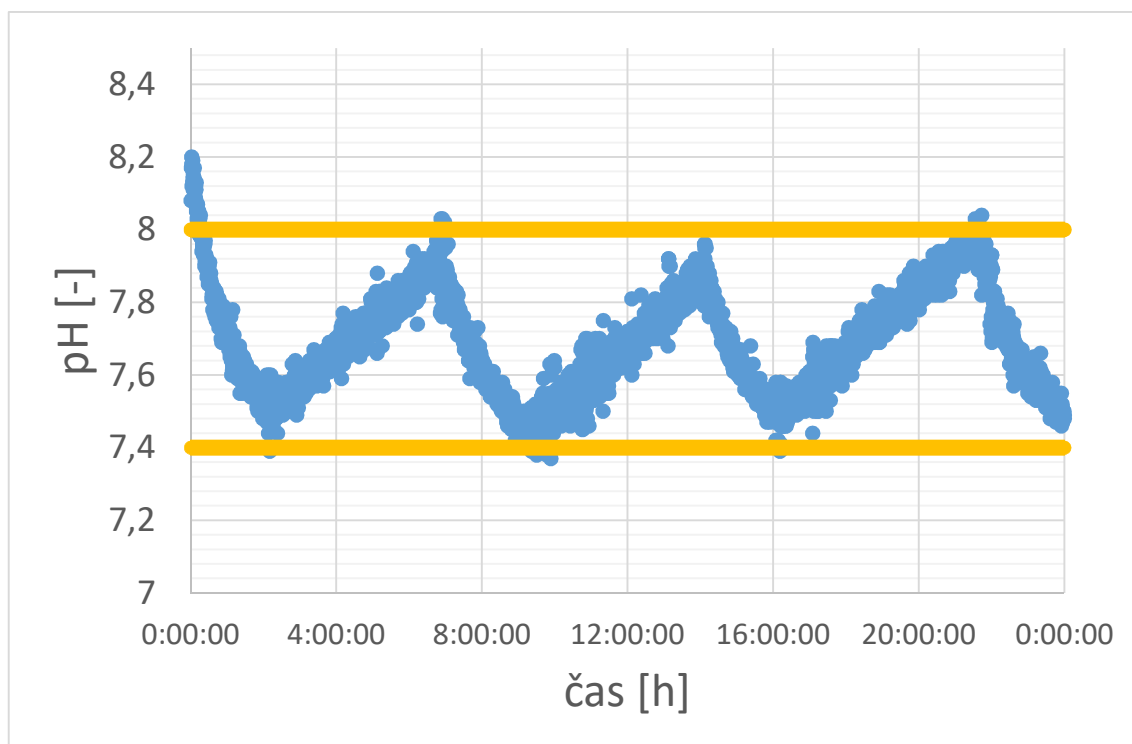


Obrázek 35: Zobrazení průběhu hodnot pH vody během dne bez dodávání CO₂

Z obrázku 35 je patrné, že hodnota pH vody v noci klesá a přes den roste. Je to dáno tím, že přes den rostliny spotřebovávají CO₂ a v noci ho produkují.

6.1.2 S dodáváním CO₂

Na obrázku 36 je zobrazen průběh hodnot pH vody v akváriu při dodávání CO₂. Jako požadovaná hodnota bylo nastaveno pH 7,7 a hystereze 0,3 jednotky pH.



Obrázek 36: Zobrazení hodnot pH vody při dodávání CO₂

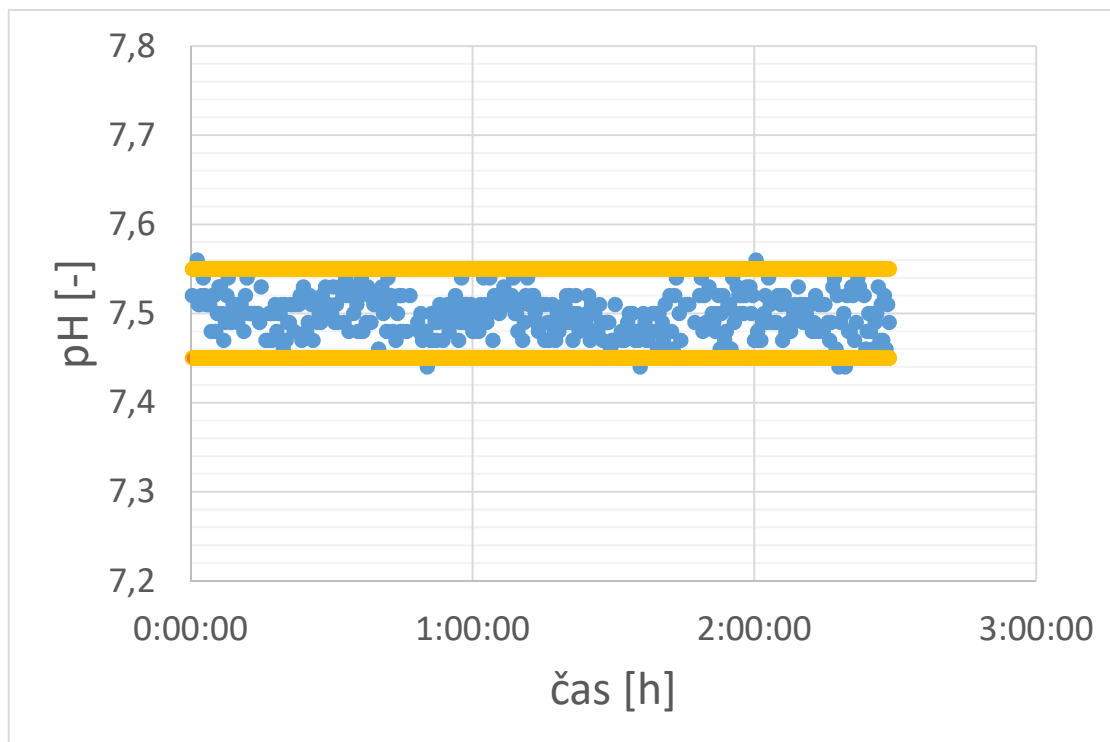
Při tomto měření bylo akvárium osvětleno nepřetržitě. K měření množství dodávaného CO₂ byl použit bublinkoměr Aqua Nova NCO2-7, podle kterého byl průtok CO₂ do akvária 270 bublinek za minutu. Ostatní podmínky byly totožné jako při měření změn hodnot pH bez dodávání CO₂.

Na obrázku jsou žlutou barvou znázorněny hranice požadovaného rozsahu hodnot pH. Před začátkem měření nebyl do akvária dodáván CO₂, proto je na začátku měření hodnota pH vody mimo vyznačený rozsah.

Z obrázku je patrné, že systém plní svoji funkci a pH vody se pohybuje ve zvoleném rozsahu. V tomto konkrétním případě je také patrné, že se hodnota pH při dodávání CO₂ snižuje přibližně dvakrát rychleji než jak stoupá pokud je dodávání CO₂ zastaveno. Toto tvrzení ovšem nelze zobecnit, protože výsledná rychlost změny pH vody závisí na objemu akvária a množství dodávaného CO₂.

6.1.3 Konstantní hodnota pH

Na obrázku 37 je zobrazen průběh hodnot pH vody při nastavení požadované hodnoty pH na 7,5 a hystereze pouze 0,05 pH. Testování probíhalo 2,5 hodiny. Ostatní podmínky jsou shodné s předchozím bodem.



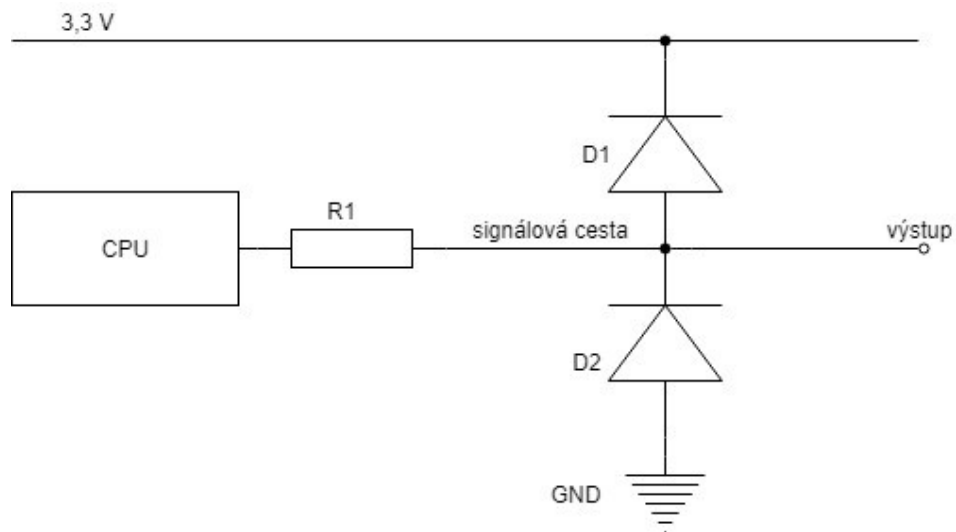
Obrázek 37: Zobrazení hodnot pH vody

Z obrázku 37 je patrné, že systém dokáže udržovat hodnotu pH v rozsahu 0,1 stupně pH. To je přibližně desetkrát přesnější než požadavek vyplývající z kapitoly 1.2

6.2 Možná vylepšení

Při testování systému se projevila chyba návrhu desky plošného spoje. Mezi piny na desce, které jsou spojeny s procesorem na i.Mx8 a tranzistory, které spínají výstupní obvody na desce plošného spoje jsou příliš dlouhé signálové cesty a jsou vedeny blízko silových cest (spínaná zásuvka). To má při spínání a rozepínání zátěže za následek indukování napětí, které by mohlo zničit procesor.

Z toho důvodu je vhodné cestu opatřit ochranou proti přepětí viz obrázek 38. Při výskytu většího napětí než je součet prahového napětí diody D1 (pro křemíkovou diodu typicky 0,6 V) a napájecího napětí (v tomto případě 3,3 V) dojde k otevření diody D1 a přepětí je přes napájecí zdroj uzemněno. Dioda D2 zabraňuje tomu, aby se na signálové cestě vyskytlo záporné napětí větší než prahové napětí diody D2. Tedy pokud se na signálové cestě vyskytne záporné napětí $|-U| > 0,6 \text{ V}$ otevře se přechod diody D2 a signálová cesta bude přes diodu uzemněna. V případě, že do procesoru poteče elektrický proud, vytvoří na rezistoru R1 úbytek napětí. Toto opatření omezí proud tekoucí do procesoru. Typická hodnota rezistoru v tomto zapojení je 100Ω . Vzhledem k faktu, že jsou spínány jen unipolární tranzistory, předpokládá se, že v běžném provozu cestou poteče jen zanedbatelně malý proud a na rezistoru R1 se nevytvoří úbytek napětí. Proto přítomnost rezistoru R1 nemá vliv na funkčnost zapojení.



Obrázek 38: Přepětová ochrana signálové cesty

6.3 Problém s i.Mx8DV

Při použití platformy i.Mx8DV se vyskytly problémy s vnitřním časovačem, které znemožnili použití této platformy pro vytvoření systému správy akvária. Po konzultaci ve firmě NXP byla platforma nahrazena vhodnější variantou i.Mx8QM. Ta plní svoji funkci bezproblémově.

7 ZÁVĚR

Cílem práce bylo navrhnout a vytvořit inteligentní systém pro správu akvária. Systém se skládá z hardwarového rozhraní a aplikace. Hardwarové rozhraní slouží k propojení čidel a výstupních obvodů s řídicí deskou i.Mx8. Aplikace zajišťuje sycení vody CO₂ na základě pH vody.

V první kapitole jsou analyzovány vhodné podmínky pro organismy v akváriu a jejich potřebná přesnost regulace. Důraz je kladen na pH vody, teplotu vody a intenzitu osvětlení. Jsou zde vyjmenovány konkrétní hodnoty teploty a pH vody pro několik vybraných organismů.

Druhá kapitola přibližuje použitou platformu i.Mx8, použitý operační systém QNX, framework Qt a vývojové prostředí Qt Creator. Jsou zde charakterizovány základní třídy Qt a popsán princip signálu a slotu.

Kapitola číslo tři se zabývá návrhem hardwarového rozhraní mezi čidly spolu s výstupními obvody a platformou i.Mx8. Jsou zde diskutovány různé způsoby měření pH vody, teploty vody a intenzity osvětlení. Dále jsou v této kapitole vybrány vhodná čidla pro snímání potřebných veličin a vhodný AD převodník.

Čtvrtá kapitola vysvětluje zejména pojmy spojené s komunikací po síti, které jsou důležité pro správné softwarové řešení vzdálené správy vytvořeného systému.

V kapitole číslo pět je popsáno celé řešení softwarové části systému pro správu akvária. Na začátku kapitoly jsou diskutovány možná řešení vzdáleného přístupu k aplikaci. Dále je představena architektura vytvořené aplikace. Následuje zobrazení nejdůležitějších signálů a slotů spolu se základními algoritmy této aplikace. V posledních dvou podkapitolách je popsáno vytvořené grafické uživatelské rozhraní a webové rozhraní pro vzdálený přístup.

Kapitola šestá vyhodnocuje naměřená data a popisuje možné zlepšení tohoto systému.

V rámci bakalářské práce byl vytvořen funkční systém pro správu akvária. Aplikace je napsána v programovacím jazyce C++ s využitím frameworku Qt. Je vytvořena pro operační systém QNX a přizpůsobena pro použití na platformě i.Mx8.

Byly splněny všechny body zadání. Nad rámec zadání byly do systému přidány funkce pro spínání topného tělesa a ovládání osvětlení na základě času nebo na základě snímané hodnoty dopadající intenzity světla. Byla přidána možnost trvale sepnout libovolný výstupní obvod. Nad rámec zadání je také zpracována jednoduchá kalibrace pH sondy a možnost využít PWM pro zapínání a vypínání LED osvětlení.

Pozornost při budoucím vývoji bude patřit převážně grafickému zpracování jak samotného GUI, tak zejména webového rozhraní.

LITERATURA

- [1] SLAVOTÍNEK, Jan. *Systém podpory akvarijních životních podmínek*. Brno, 2013. Bakalářská práce. VUT FEKT. Vedoucí práce Ing. Petr Petiovský.
- [2] *Rybicky.net* [online]. [cit. 2017-10-30]. Dostupné z: www.rybicky.net
- [3] *Aquarent.info* [online]. [cit. 2017-11-13]. Dostupné z: <https://www.aquarent.info/suesswasser/aquarium-technik/aquarienheizer/5226/eheim-thermopreset-aquariumheizer>
- [4] *Chovatelské potřeby: akvaristika a teraristika* [online]. [cit. 2017-11-12]. Dostupné z: <http://www.migashop.cz/miga/eshop/4-1-Akvaristika/1553-4-Vyhrev-dna-akvaria/5/14610-DENNERLE-TOPNY-KABEL-ECO-LINE-12V-20W-120-200-L>
- [5] *Dennerle* [online]. [cit. 2017-11-02]. Dostupné z: <http://dennerle.com/cs/poradce/akvaristika/hnojeni-co2/proc-dodavat-do-akvaria-co2>
- [6] Operační systém. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-11-12]. Dostupné z: https://cs.wikipedia.org/wiki/Operační_systém
- [7] DOŠEK, Bc. Roman. *OS Linux na platformě ARM*. Zlín, 2013. Diplomová práce. Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky. Vedoucí práce Ing. Tomáš Dulík, Ph.D.
- [8] *Wikipedie* [online]. [cit. 2017-10-30]. Dostupné z: https://cs.wikipedia.org/wiki/Operační_systém_reálného_času
- [9] *Wikipedie* [online]. [cit. 2017-11-02]. Dostupné z: <https://cs.wikipedia.org/wiki/Mikrojádru>
- [10] *Wikipedie* [online]. [cit. 2017-11-02]. Dostupné z: https://cs.wikipedia.org/wiki/Monolitické_jádro
- [11] *Bogdangradinaru* [online]. [cit. 2017-10-30]. Dostupné z: <https://bogdangradinaru.wordpress.com/2010/02/04/kernel-vs-operating-system/>
- [12] *Nullpwd.wordpress.com* [online]. [cit. 2017-11-12]. Dostupné z: <https://nullpwd.wordpress.com/2011/04/08/want-to-write-an-operating-system/>
- [13] SVOBODA, František. *Implementace vybraných metod pro plánování úloh v RTOS μ C/OS-II*. Plzeň, 2012. Diplomová práce. Západočeská univerzita v Plzni
- [14] *Arduino shop* [online]. [cit. 2017-11-21]. Dostupné z: <https://arduino-shop.cz/arduino-vstupni-periferie/1250-tekute-ph-hodnota-detekce-detekovat-senzor-modul-monitorovani-ovladani-pro-arduino-1459105294.html>
- [15] ROKYTA, Michal. *Studium možností využití SOC systémů* [online]. Pardubice, 2010 [cit. 2017-12-05]. Bakalářská práce. Univerzita Pardubice, Fakulta elektroniky a informatiky. Vedoucí práce Ing. Radim Vondra, Ph.D.
- [16] HORÁK, Martin. *Bezdrátový mesh protokol pro FreeRTOS* [online]. Brno, 2014 [cit. 2017-12-05]. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Vedoucí práce Ing. Vladimír Červenka.
- [17] JEŘÁBEK, Lukáš. *Analyzátor vodného roztoku* [online]. Brno, 2009 [cit. 2017-12-05]. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Vedoucí práce Ing. Pavel Šteffan, Ph.D.
- [18] *Datasheet fotorezistoru* [online]. [cit. 2017-12-05]. Dostupné z: <http://akizukidenshi.com/download/ds/senba/GL55%20Series%20Photoresistor.pdf>

- [19] *Technické měření* [online]. Praha [cit. 2017-12-06]. Dostupné z: <http://www1.fs.cvut.cz/cz/U12110/TEM/senzory/grafy.htm>. ČVUT Praha.
- [20] *World of Sensors* [online]. [cit. 2017-12-10]. Dostupné z: <http://sensorxyz.blogspot.cz/2016/03/lm35-precision-centrigrade-temperature.html>
- [21] Texas Instruments. *Ti.com: datasheet* [online]. [cit. 2017-12-10]. Dostupné z: <http://www.ti.com/lit/ds/symlink/lm35.pdf>
- [22] REK, Tomáš. *Analyzátor AD převodníků* [online]. Pardubice, 2009 [cit. 2017-12-10]. Dostupné z: http://dspace.upce.cz/bitstream/handle/10195/34869/RekT_Analyzator_AD_MH_2009.pdf?sequence=1. Bakalářská práce. Univerzita Pardubice, Fakulta elektrotechniky a informatiky. Vedoucí práce Ing. Martin Hájek.
- [23] VONÁŠEK, Jiří. *Návrh a realizace A/D převodníku s jednočipovým mikropočítačem* [online]. Blatná, 2007 [cit. 2017-12-10]. Dostupné z: https://theses.cz/id/nu3gm3/downloadPraceContent_adipIdno_6382. Bakalářská práce. Jihočeská univerzita v Českých Budějovicích. Vedoucí práce PaedDr. Petr Adámek, Ph.D.
- [24] STARÝ, Richard. *Decimace výstupního datového toku $\Sigma\Delta$ modulátoru - 1. sekce* [online]. Praha [cit. 2017-12-10]. Dostupné z: <http://richardstary.wz.cz/fel/8/psp/sem/zprava.pdf>. Semestrální práce. České vysoké učení technické v Praze, Fakulta elektrotechnická.
- [25] BLAHA, Petr a Petr VAVŘÍNEK. *Řízení a regulace 1: Základy regulace lineárních systémů* [online]. Brno, 2009 [cit. 2018-03-25]. Dostupné z: <https://www.vutbr.cz/studenti/predmety/detail/184071>. Skripta. FEKT VUT.
- [26] OBDŘÍŽÁLEK, Jan. *Nernstova rovnice srozumitelně* [online]. 2003 [cit. 2018-03-26]. Dostupné z: <http://utf.mff.cuni.cz/~jobdr/download/NERNST.pdf>
- [27] *Arduino shop: PH modul pro Arduino* [online]. [cit. 2018-03-26]. Dostupné z: https://arduino-shop.cz/arduino-vstupni-periferie/1250-tekute-ph-hodnota-detekce-detekovat-senzor-modul-monitorovani-ovladani-pro-arduino-1459105294.html?gclid=EAIaIQobChMI-6qs98KK2gIVDbobCh0e9ATtEAQYASABEgIyOPD_BwE
- [28] *Mall.cz: pH metry* [online]. [cit. 2018-04-04]. Dostupné z: https://www.mall.cz/ph-metry/voltcraft-ph-100-atc?gclid=EAIaIQobChMInfyErIsh2gIV5LftCh1pxgDbEAQYASABEgIVHPD_BwE
- [29] *Sklorex-akvarium.cz* [online]. [cit. 2018-04-04]. Dostupné z: <http://www.sklorex-akvarium.cz/p3037-co2-set-aqr-bez-nocniho-vypinani-lahev-2-l.html>
- [30] Qt (knihovna). In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2018-04-08]. Dostupné z: [https://cs.wikipedia.org/wiki/Qt_\(knihovna\)](https://cs.wikipedia.org/wiki/Qt_(knihovna))
- [31] CHROBOCZEK, Martin. *Grafická uživatelská rozhraní v Qt a C: [plně kompatibilní s Qt 5]*. Brno: Computer Press, 2013. ISBN 978-80-251-4124-3.
- [32] DOČKAL, Michal. *SW PRO OBSLUHU TESTŮ A GENEROVÁNÍ PROTOKOLŮ* [online]. Brno, 2012 [cit. 2018-04-08]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=67646. Bakalářská práce. VUT FEKT. Vedoucí práce Ing. Lešek Franek.
- [33] *Tizen* [online]. [cit. 2018-04-08]. Dostupné z: <https://developer.tizen.org/development/guides/native-application/user-interface/dali/event-handling>

- [34] JEŽEK, Svatopluk. *Diagnostika síťového provozu v síti LAN v prostředí OS Windows v konkrétní firmě* [online]. Praha, 2010 [cit. 2018-05-13]. Dostupné z: <http://info.sks.cz/www/zavprace/soubory/71900.pdf>. Bakalářská práce. Vysoká škola ekonomická v Praze, Fakulta informatiky a statistiky.
- [35] PREČAN, David. *DÁLKOVÉ ŘÍZENÍ ZAPÍNÁNÍ INFORMAČNÍHO SYSTÉMU – ČÁST II* [online]. BRno, 2018 [cit. 2018-05-13]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=8779. Bakalářská práce. VUT v Brně, fakulta elektrotechniky a komunikačních technologií, ústav telekomunikací. Vedoucí práce Ing. IVO HERMAN, CSc.
- [36] ŠŤASTNÝ, Petr. *Monitoring DNS serverů domén druhé úrovně* [online]. Praha, 2010 [cit. 2018-05-14]. Dostupné z: <https://is.cuni.cz/webapps/zzp/download/120009478/?lang=cs>. Diplomová práce. Univerzita Karlova v Praze, Matematicko-fyzikální fakulta. Vedoucí práce RNDr. Ing. Jiří Peterka.
- [37] ŠIMEK, Pavel. *Návrh dynamických internetových stránek a implementace xml* [online]. Brno, 2008 [cit. 2018-05-14]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=7759. Bakalářská práce. Vysoké učení technické v Brně, fakulta podnikatelská. Vedoucí práce Ing. Jiří Kříž, Ph.D.
- [38] Instructables. *Instructables.com* [online]. 2015 [cit. 2018-05-15]. Dostupné z: <http://www.instructables.com/id/Control-an-LED-with-PWM-output/>
- [39] KVASNICA, Josef. *Síťová asynchronní aplikace Piškvorky* [online]. Jihlava, 2012 [cit. 2018-05-18]. Dostupné z: <https://is.vspj.cz/bp/get-bp/student/29408/thema/3234>. Bakalářská práce. Vysoká škola polytechnická Jihlava. Vedoucí práce PaedDr. František Smrčka, Ph.D.

Seznam příloh

Příloha A.1 – Schema obvodového zapojení

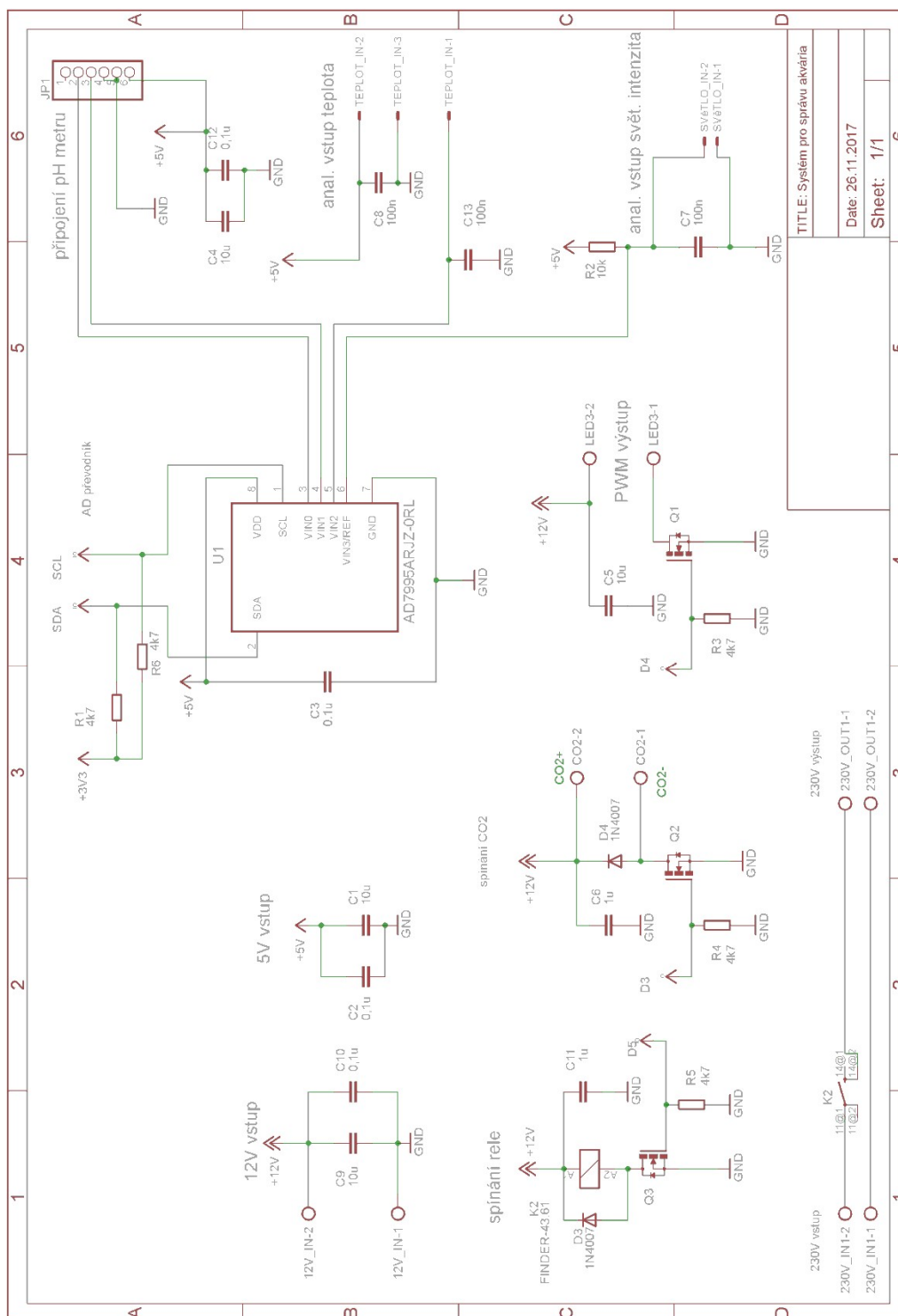
Příloha A.2 – Zobrazení desky pološného spoje

Příloha B – Seznam součástek

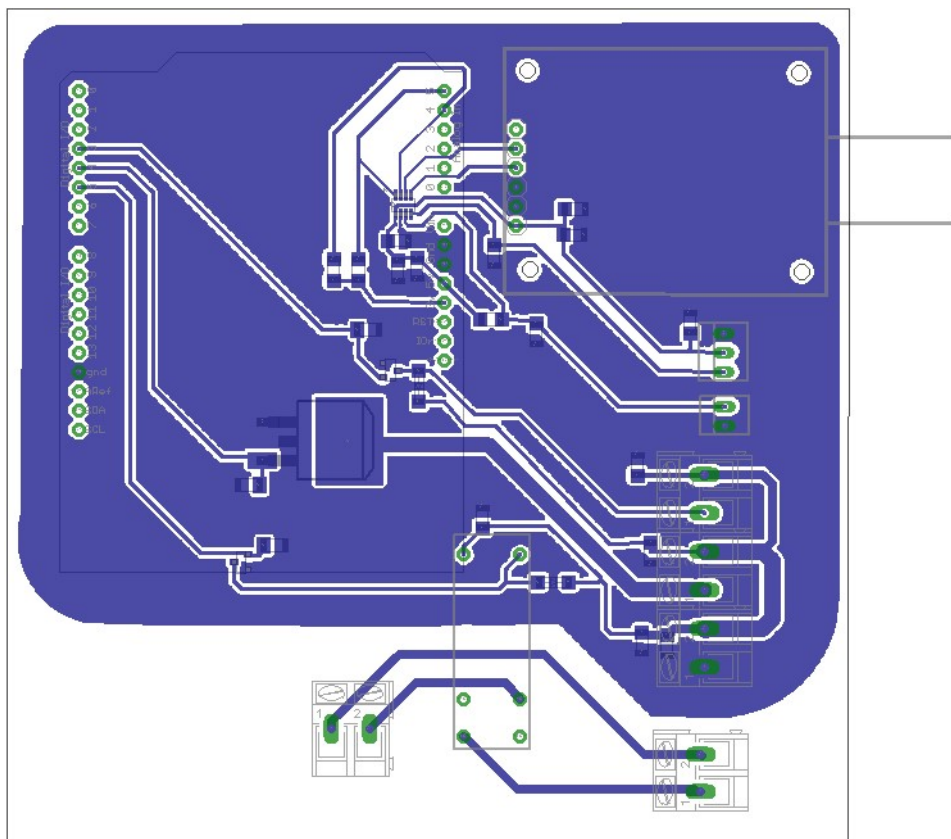
Příloha C – Zdrojové kódy aplikace jsou uloženy na přiloženém CD

A NÁVRH ZAŘÍZENÍ

A.1 Obvodové zapojení



A.2 Deska plošného spoje řídicího modulu – bottom (strana spojů)



Rozměr desky 112 x 110 [mm], měřítko M1:1

B SEZNAM SOUČÁSTEK

Označení	Hodnota	Pouzdro	Popis
12V_IN		AK300/2	Konektor
230V_IN1		AK300/2	Konektor
230V_OUT1		AK300/2	Konektor
C1	10u	C1206	Keramický kondenzátor
C2	0,1u	C1206	Keramický kondenzátor
C3	0,1u	C1206	Keramický kondenzátor
C4	10u	C1206	Keramický kondenzátor
C5	10u	C1206	Keramický kondenzátor
C6	1u	C1206	Keramický kondenzátor
C7	100n	C1206	Keramický kondenzátor
C8	100n	C1206	Keramický kondenzátor
C9	10u	C1206	Keramický kondenzátor
C10	0,1u	C1206	Keramický kondenzátor
C11	1u	C1206	Keramický kondenzátor
C12	0,1u	C1206	Keramický kondenzátor
C13	100n	C1206	Keramický kondenzátor
CO2		AK300/2	Konektor
D3	1N4007	SMA	Dioda 1N4007
D4	1N4007	SMA	Dioda 1N4007
JP1		1X06	Pin header pro modul pH
K2		FINDER-43.61	Spínací relé
LED3		AK300/2	Konektor
Q1	AOD484	D2PACK	Unipolární tranzistor N-kanál
Q2	IRLML2402PBF	MICRO3	Unipolární tranzistor N-kanál
Q3	IRLML2402PBF	MICRO3	Unipolární tranzistor N-kanál
R1	4k7	C1206	Rezistor
R2	10k	C1206	Rezistor
R3	4k7	C1206	Rezistor
R4	4k7	C1206	Rezistor
R5	4k7	C1206	Rezistor
R6	4k7	C1206	Rezistor
Světlo_IN		1X02	Pin header
TEPLOT_IN		1X02	Pin header
U1	AD7995	SOT23	ADC